# Dependable Network Protocols in Wireless Sensor Networks

**Tiong Hoo Lim**

PhD

University of York
Department of Computer Science

**September 2013**

# Abstract

This thesis is concerned with the dependability of Wireless Sensor Networks (WSNs). We propose an approach, inspired by the immune system, that allows individual nodes to detect, diagnose and recover from different failures by switching between different protocols using a multi-modal switching mechanism. A causal link between different failures in WSN is identified. Existing fault tolerance in WSNs approaches are examined. From the survey, it is identified that various attempts have been made to improve the fault tolerance of the communication protocol especially in the routing protocols. Although tests have been performed to evaluate the communication protocols prior to deployment, failures in WSNs are still being reported when deployed in real environments. A Systematic Protocol Evaluation Technique (SPET) is proposed and applied to evaluate the dependability of the proposed multi-modal protocol and reduce the uncertainties in the experiment and to demonstrate the confidence in the measurements taken from experiments.

# Contents


**1 Introduction** **1**

1.1 Motivation . . . . . 1

1.1.1 Wireless Sensor Networks . . . . . 2

1.1.2 Dependable Wireless Sensor Networks . . . . . 2

1.1.3 Effect of Radio Abnormality on the Dependability of Wireless Sensor Networks . . . . . 3

1.1.4 Dependable Routing . . . . . 3

1.1.5 Fault Tolerance Approach toward Dependability . . . . . 4

1.1.6 Establishing Dependability with Confidence . . . . . 5

1.2 Statement of Hypothesis . . . . . 6

1.3 Contributions . . . . . 6

1.4 Organisation of Thesis . . . . . 7

**2 Towards a Dependable Wireless Sensor Networks** **9**

2.1 Wireless Sensor Networks . . . . . 9

2.2 Design Challenges in WSNs . . . . . 14

2.2.1 Tolerate failure . . . . . 15

2.2.2 Efficiency . . . . . 15

2.2.3 Scalability . . . . . 16

2.2.4 Adaptability . . . . . 16

2.2.5 Discussion . . . . . 16

2.3 Dependable WSN . . . . . 17

2.4 Failures in WSNs . . . . . 17

2.4.1 Fault, Error and Failure . . . . . 18

2.4.2 Fault Characteristics . . . . . 19

2.5 Type of Anomalies . . . . . 20

2.5.1 Data anomalies . . . . . 20

2.5.2 Node anomalies . . . . . 21

2.5.3 Network anomalies . . . 22
2.5.4 Signal anomalies . . . 22
2.5.5 Discussion . . . 23
2.6 Achieving Dependable Network using Fault Tolerant Approaches . 23
2.6.1 Fault tolerant architecture . . . 25
2.6.2 Error Detection . . . 27
2.6.3 Error Consensus . . . 32
2.6.4 Error Diagnosis . . . 33
2.6.5 Error Recovery . . . 34
2.6.6 Limitation of Existing Fault Tolerant Approaches . . . 37
2.7 The Application of Immune-Inspired Approaches toward Fault Tolerance . . . 38
2.7.1 The Immune System . . . 39
2.7.2 Immunological Theory . . . 41
2.7.3 Artificial Immune System . . . 43
2.7.4 Application of AIS in WSNs . . . 46
2.7.5 AIS for Interference Recognition in WSNs . . . 47
2.8 Fault Tolerance in WSNs Routing Protocols . . . 48
2.8.1 Reactive Routing in WSNs . . . 48
2.8.2 Mixed Routing . . . 49
2.8.3 Research Attempts to Enhance and Augment AODV . . . 50
2.9 Current Evaluation methodology . . . 54
2.9.1 The hybrid approaches . . . 56
2.9.2 Reality Gap in the Current Evaluation Approaches . . . 56
2.10 Summary . . . 57

**3 Evaluating the Fault Tolerance of the WSNs Routing Protocols 59**
3.1 Materials and Methods . . . 59
3.1.1 Application Environment and Topology . . . 60
3.1.2 Failure Model . . . 62
3.1.3 Experimental Tools . . . 65
3.1.4 Simulation Methodology . . . 66
3.1.5 Evaluation Metrics . . . 67
3.1.6 Statistical tools . . . 68
3.2 Performance Analysis of Routing Protocols . . . 71
3.2.1 Observations . . . 72
3.2.2 Discussion . . . 94
3.3 Summary . . . 94

**4 Achieving Dependability using Multi-modal Network Protocol 95**
4.1 Motivation . . . 95
4.2 Multi-modal Protocol . . . 96
4.2.1 Multimodal Routing Protocol . . . 96
4.2.2 Design of Multi-modal Routing Protocol . . . 97
4.2.3 Discussion . . . 101
4.3 On Robustness: 1. The effects of failure duration . . . 102
4.3.1 Application Scenario . . . 102
4.3.2 Network Setup . . . 102
4.3.3 Simulation Parameters . . . 103
4.3.4 Simulation of failures . . . 104
4.3.5 Results . . . 105
4.4 On Robustness: 2. The effects of varying the number of failing nodes 111
4.4.1 Results . . . 111
4.4.2 Discussion on the effect of failure size . . . 131
4.5 On Scalability: The effects of network size . . . 136
4.5.1 Packet Generation . . . 136
4.5.2 Fault injection . . . 137
4.5.3 Results . . . 138
4.5.4 Discussion on the scalability of MRP . . . 143
4.6 Summary . . . 143

**5 Assisted Recovery with An Immune-Inspired Classifier 144**
5.1 Motivation . . . 145
5.1.1 Problem Formulation . . . 146
5.1.2 Existing approaches . . . 147
5.1.3 Contributions . . . 148
5.2 The RDA: Receptor Density Algorithm . . . 149
5.2.1 Biological principle . . . 149
5.2.2 The Algorithm . . . 150
5.3 IDRS: Interference Detection and Recovery Systems . . . 151
5.3.1 MDM: The MRP Detection Module . . . 153
5.3.2 RDM: The RDA Diagnostic Module . . . 153
5.3.3 RIRM: Radio Interference Response Module . . . 159
5.4 Experiments and Results . . . 160
5.4.1 Evaluation of the RDM . . . 160
5.4.2 Evaluation of the IDRS . . . 164
5.4.3 Discussion . . . 177
5.5 Summary . . . 177

**6 Using Statistical Approach to Demonstrate Dependability** **179**
6.1 Motivation . . . 179
6.2 Scientific Protocol Evaluation Technique . . . 180
6.2.1 Defining the Scope . . . 181
6.2.2 Design of Experiment . . . 182
6.2.3 Uncertainty Analysis . . . 182
6.2.4 Graphical Analysis . . . 182
6.2.5 Statistical Analysis . . . 183
6.2.6 Measuring the Effectiveness of the WSN's Evaluation . . . 185
6.3 Case Study: Evaluating the dependability of MRP using SPET . . . 187
6.3.1 Defining the Objectives . . . 187
6.3.2 Experimental Setup . . . 188
6.3.3 Results Analysis . . . 191
6.3.4 Discussion: Benefit of SPET . . . 201
6.4 Summary . . . 204

**7 Conclusions and Future Works** **205**
7.1 Significant Contributions of the studies . . . 205
7.2 Revisiting the research questions . . . 207
7.3 Recommendations for further research . . . 209

**A Screenshoot for Grid Topology** **211**

**Abbreviation** **216**

**Glossary** **218**

**Bibliography** **221**

# List of Tables


2.1 Comparison between different fault tolerant architectures . . . . . 27
2.2 Comparison of various anomaly detection algorithms . . . . . 30
2.3 Route detection and recovery mechanisms in different variants of AODV routing protocol. . . . . 53
2.4 Survey from 50 papers published in IEEE MASS 2010-2012 . . . . . 55

3.1 The values of ON/OFF periods representing short, medium and long interferences. . . . . 64
3.2 Generic NS-2 Parameters used in the work . . . . . 67
3.3 Normality Test for the simulated results . . . . . 70
3.4 The range of *A*-values representing different effect sizes. . . . . 71
3.5 *p*-values computed from Rank-Sum test for different PDR with 0.1s failure duration . . . . . 74
3.6 *p*-values computed from Rank-Sum test for different PDR with 10s failure durations . . . . . 75
3.7 *A*-values computed from Vargha-Delaney Test for different PDR with 0.1s failure duration . . . . . 76
3.8 *A*-values computed from Vargha-Delaney Test for different PDR with 10s failure durations . . . . . 77
3.9 *p*-values computed from Rank-Sum test for different energy utilisation with 0.1s failure duration . . . . . 79
3.10 *p*-values computed from Rank-Sum test for different energy utilisation with 10s failure durations . . . . . 80
3.11 *A*-values computed from Vargha-Delaney Test for different energy utilisation with 0.1s failure durations . . . . . 81
3.12 *A*-values computed from Vargha-Delaney Test for different energy utilisation with 10s failure durations . . . . . 82

3.13 The total number of MAC packets transmitted and received are higher in AOMDV compared to AODV, NST and TinyAODV. . . . 82
3.14 *p*-values computed from Rank-Sum test for different routing overhead with 0.1s failure duration . . . . . 85
3.15 *p*-values computed from Rank-Sum test for different routing overhead with 10s failure durations . . . . . 86
3.16 *A*-values computed from Vargha-Delaney Test for different routing overhead with 0.1s failure duration . . . . . 87
3.17 *A*-values computed from Vargha-Delaney Test for different routing overhead with 10s failure duration . . . . . 88
3.18 *p*-values computed from Rank-Sum test for different average packet delay with 0.1s failure duration . . . . . 90
3.19 *p*-values computed from Rank-Sum test for different average packet delay with 10s failure duration . . . . . 91
3.20 *A*-values computed from Vargha-Delaney Test for different average delay with 0.1s failure duration . . . . . 92
3.21 *A*-values computed from Vargha-Delaney Test for different average delay with 10 failure durations . . . . . 93

4.1 NS-2 Parameters to evaluate MRP. . . . . 104
4.2 NS-2 traces showing the packet dropped due to collision. . . . . 105
4.3 The mean, median for different failure durations with 10 failing nodes. . . . . 108
4.4 *p*- and *A*-values between MRP and AODV, MRP and NST, and NST and AODV for different failure durations . . . . . 109
4.5 The median of the PDR for different number of failures injected with different failure durations for MRP, NST, and AODV. . . . . 113
4.6 The median of the average energy remaining in a node for different number of failures injected with different failure durations for MRP, NST, and AODV. . . . . 118
4.7 The median of the normalised routing overhead for different number of failures injected with different failure durations for MRP, NST, and AODV. . . . . 125
4.8 The median of the average end to end delay for each packet for different failure durations for MRP, NST, and AODV. . . . . 130
4.9 The summary of the traffic flow patterns for different network sizes. 137
4.10 The table showing the median and mean of the PDR, energy remains, routing overhead and average delay for the network size of 25-, 49-, 100-, 225, 400, and 900-nodes. . . . . 139

4.11 $p$- and $A$-values between MRP and AODV, MRP and NST, and NST and AODV for different network sizes . . . . . 142

5.1 Interference Class based on the intensity and duration . . . . . 163
5.2 Descriptions of different type of traffics generated from the laptop. 164
5.3 The execution of different responses in all the nodes for IDRS and MTPC . . . . . 167
5.4 $p$-values of the Wilcoxon rank sum test to determine statistical significance of the performance between the routing protocols. . . . . 168
5.5 Vargha-delaney test to determine scientific significance of the performance between the routing protocols . . . . . 169
5.6 Hardware requirements in term of computational and memory footprint introduced by MRP and IDRS . . . . . 169
5.7 $p$ and $A$-values for AODV, NST and MRP against IDRS for the performance metrics PDR, ENG, RT and DLY. . . . . 176

6.1 $p$ and $A$ values for Hardware Experiment (Bold highlights significance value) . . . . . 194
6.2 $p$ and $A$ values for Simulation (Bold highlights significance value) . 195
6.3 KS Test $p$-values (Bold indicates two samples having the same distribution) for PDR between NS2 and TinyOS using Matlab . . . . . 196
6.4 $p$ and $A$ values for Hardware Experiment (Bold highlights significance value) . . . . . 199
6.5 $p$ and $A$ values for Simulation (Bold highlights significance value) . 200
6.6 KS Test $p$-values showing the routing overhead similarity between NS2 and TinyOS using Matlab . . . . . 200
6.7 This table presents the frequency (in $\%$) where one protocol is better ($>$) than the other over 500 runs as the sample size increases. . . 202

# List of Figures


2.1 WSNs deployment showing the single-hop (left) and multihop network (right). In a multihop network, a routing protocol is required. 10
2.2 An example of WSNs Hardware . . . . . 11
2.3 Frequency Allocation for WSNs starts from channel 11 to 26. . . . . 12
2.4 Formation and manifestation mechanisms of faults, errors, and failures . . . . . 18
2.5 Examples of manifestation of faults caused by communication fault in WSNs . . . . . 19
2.6 The causality between signal, network, node and application anomalies . . . . . 24

3.1 An Experimental Framework based on the existing State of the Art experimental techniques in WSNs . . . . . 60
3.2 An outdoor deployment based on 7 by 7 grid topology . . . . . 61
3.3 A typical WLAN traffic patterns. . . . . 63
3.4 The patterns showing the number of successful packets received by a node during WLAN interference . . . . . 64
3.5 The PDR of NST is higher than AODV, AOMDV and TinyAODV as the number of failing nodes increases. . . . . 73
3.6 Lower energy consumption is observed in NST-AODV compared to AODV due to retransmission. . . . . 78
3.7 The routing overhead generated during the evaluation of AODV, NST, AOMDV and TinyAODV for different failures. . . . . 84
3.8 The average packet delay introduced for AODV, NST, AOMDV and TinyAODV for different failures. . . . . 89

4.1 The Architecture of Multimodal Routing Protocol . . . . . 98
4.2 Flow Diagram for MRP during failures. . . . . 100

4.3 Network topology based on the indoor deployment for critical health monitoring networks . . . 103
4.4 Box-Whiskers plot showing the PDR and energy consumption of MRP against AODV and NST for different failure durations . . . 106
4.5 Box-Whiskers plot showing the delay and routing overhead of MRP against AODV and NST for different failure durations . . . 107
4.6 $p$-values for the PDR with different number of failures with 0.5s and 2s failure durations . . . 114
4.7 $p$-values for the PDR with different number of failures with 10s and 20s failure durations . . . 115
4.8 $A$-values for the PDR with different number of failures with 0.5s and 2s failure durations . . . 116
4.9 $A$-values for the PDR with different number of failures with 10s and 20s failure durations . . . 117
4.10 $p$-values for the energy remain with different number of failures with 0.5s and 2s failure durations . . . 119
4.11 $p$-values for the energy remain with different number of failures with 10s and 20s failure durations . . . 120
4.12 $A$-values for the energy remain with different number of failures with 0.5s and 2s failure durations . . . 121
4.13 $A$-values for the energy remain with different number of failures with 10s and 20s failure durations . . . 122
4.14 $p$-values of the normalised routing overhead for different number of failures with 0.5s and 2s failure durations . . . 126
4.15 $p$-values of the normalised routing overhead for different number of failures with 10s and 20s failure durations . . . 127
4.16 $A$-value of the normalised routing overhead for different number of failures with 0.5s and 2s failure durations . . . 128
4.17 $A$-value of the normalised routing overhead for different number of failures with 10s and 20s failure durations . . . 129
4.18 $p$-values for average end-to-end delay with different number of failures with 0.5s and 2s failure durations . . . 132
4.19 $p$-values for average end-to-end delay with different number of failures with 10s and 20s failure durations . . . 133
4.20 $A$-values for the average end-to-end delay with 0.5s and 2s failure durations . . . 134
4.21 $A$-values for the average end-to-end delay with 10s and 20s failure durations . . . 135

4.22 Network topology showing a square grid of 7 by 7. . . . . 137
4.23 Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for different network sizes . . . . . 140
4.24 Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for different network sizes . . . . . 141

5.1 Interference source is introduced near node 5 to disrupt the radio communication between node 2 and 3. . . . . 147
5.2 The behaviours of a receptor when it interacts with different signals. 151
5.3 The architecture of the Interference Detection and Recovery System. 152
5.4 The raw RSSI data collected from the radio interface of a TelosB . . 154
5.5 The signatures generated by the RDA . . . . . 156
5.6 The signature generated by RDA with the different values h . . . . 157
5.7 The signature patterns generated by RDA with the different values $\beta$ 157
5.8 Using the outputs generated by RDA, the interference can be classified into either Class I, II, or III . . . . . 158
5.9 Decision tree used by the RIRM and MDM to respond to different interference. . . . . 160
5.10 The distribution of the interference characteristics after processed by the RDA. . . . . 162
5.11 Results showing the detection accuracy and the effect of different interference classes on PSR . . . . . 165
5.12 PDR achieved by different routing protocols for different classes of interferences . . . . . 167
5.13 Transmission Overhead for hardware experiment . . . . . 168
5.14 Box-Whisker plot for the Packet Delivery Rate (%) generated by AODV, NST, MRP and IDRS. . . . . 172
5.15 Box-Whisker plot with Median and Inter-quartile range for the routing overhead (%) generated by AODV, NST, MRP and IDRS. . . . . 173
5.16 Box-Whisker plot for the energy consumed by AODV, NST, MRP and IDRS during interference . . . . . 174
5.17 Box-Whisker plot for the average delay generated by the AODV, NST, MRP and IDRS during interference. . . . . 175

6.1 Scientific Protocol Evaluation Technique to reduce the experimental uncertainty and improve the confidence of a protocol evaluation. 181

6.2 Conceptual Statistical Test Framework applies non-parametric tests to compare the distributions of the results obtained from the experiments and simulations . . . . . 184
6.3 TelosB network setup using a small network for a better control of the experiment environment . . . . . 188
6.4 The *A*-values for different sample size taken from two set of experimental results (Hardware and Simulator) . . . . . 190
6.5 Box-Whisker plot with Median and Inter-quartile range for PDR for different sample sizes and failure durations . . . . . 192
6.6 Mean and the Error Bar for PDR taken from different sample sizes with different failure durations . . . . . 193
6.7 Box-Whisker plot with Median and Inter-quartile range for RO taken from different sample sizes and failure durations . . . . . 197
6.8 Mean and the Error Bar for RO taken from different sample sizes with different failure durations . . . . . 198

A.1 Network Topology 5 by 5 (Figure A.1(a)) and 7 by 7 (Figure A.1(b) 211
A.2 Network Topology 10 by 10 . . . . . 212
A.3 Network Topology 15 by 15 . . . . . 213
A.4 Network Topology 20 by 20 . . . . . 214
A.5 Network Topology 30 by 30 . . . . . 215

# Acknowledgements


- To my parent, without your love, teaching and guidance, I will not be who I am today.
- To my beloved wife, Memeriahneh Loo for always being here for me throughout these years, enduring the good and the bad times.
- To my two lovely kids, Aby Wan Hua Lim and Amei Pei Ying Lim for being lovely and understanding children during my absence and cheering me up during my presence.
- To Jon and Iain for taking all the efforts to supervise me throughout the PhD. It has been a great pleasure working with you both.
- To my siblings, close families and relatives for the support and advise.
- To my friends for the friendly encouragement and always be there to listen to me complaining. All the joy and fun will never be forgotten.
- To the Government of Brunei, Institut Teknologi Brunei, and Brunei High Commissioner in United Kingdom for the financial support and logistics.
- To the staffs and colleagues in Computer Science Department, Real Time Systems Group (RTS), Non-Standard Computational Group (NSC), and York Centre for Complex Systems Analysis (YCCSA) for the technical support, knowledge and discussion poised throughout these four years.
- Finally, to the examiners, Prof Uwe Aickelin of University of Nottingham and Dr Leandro Soares Indrusiak of University of York for the interesting discussions and helpful comments, time and efforts during the *viva voce.*

# Declaration

The work in this thesis has been carried out by the author between December 2009 and November 2013 at the Department of Computer Science, University of York. Apart from work whose authors are clearly acknowledged and referenced, all other works presented in this thesis were carried out by the author. The results of this work have been previously published or submitted for publication by the author. A complete list of refereed publications is as follows:

- T. H. Lim, I. Bate, and J. Timmis. Multi-modal routing to tolerate failures. In *Proceedings of the 7th International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, pages 211−216. IEEE, 2011.

- T. H. Lim, H. K. Lau, J. Timmis, and I. Bate. Immune-inspired self healing in wireless sensor networks. In C. Coello, J. Greensmith, N. Krasnogor, P. Li, and M. Nicosia, G.and Pavone, editors, *Artificial Immune Systems*, volume 7597 of Lecture Notes in Computer Science, pages 42−56. Springer Berlin Heidelberg, 2012.

- T. H. Lim, I. Bate, and J. Timmis. Validation of performance data using experimental verification process in wireless sensor network. *In Proceedings of the 16th Conference on Emerging Technologies Factory Automation*. IEEE, 2012.

- T. H. Lim, I. Bate, and J. Timmis. Systematic experimental analysis and evaluation of routing protocol in wireless sensor networks. In *Proceedings of the 2012 UK Electronics Forum*. IET, 2012.

- T. H. Lim, I. Bate, and J. Timmis. Self-adaptive fault tolerance systems for a dependable Wireless Sensor Network, Special Issue on *Self-Adaptive Networked Embedded Systems* in Design Automation for Embedded Systems. Springer Journal, 2013. [To be published]

Beside the refereed publications, part of the literature reviews of the thesis are taken from the author's qualifying dissertation and progress report submitted to the department during year 1 and 2 of the PhD.

- T. H. Lim. Detecting anomalies in Wireless Sensor Networks. *Qualifying Dissertation*. University of York, 2010
- T. H. Lim. Managing Transient failures in Wireless Sensor Networks. *Progress Report*. University of York, 2011

CHAPTER

# ONE

# Introduction

## 1.1 Motivation

Wireless Sensor Networks (WSNs) consist of a number of wireless sensor nodes that are deployed across an area of interest to perform surveillance and monitoring tasks (Akyildiz and Vuran, 2010). They have become a popular area of research in recent years due to their huge potential to be used in various applications. Ever since the first smart dust military project proposed by Kahn et al. (1999) using small communication device, different application areas have emerged including wildlife monitoring (Mainwaring et al., 2002; Anthony et al., 2012), plantation monitoring (Langendoen et al., 2006), health-care (Malan et al., 2004), fire detection systems (Liu et al., 2013), and military operation (Lédeczi et al., 2005). With the adoption of WSNs in safety critical systems, it is not only sufficient for WSNs to maintain operating for a long period of time with best effort delivery service. They must provide a reliable end-to-end communication for the application to maintain operation. Despite various success stories of WSNs in test environments, evidence from previous deployments have shown unacceptable levels of reliability (Shakshuki et al., 2009). With limited and irreplaceable energy resources available in the node, nodes can fail due to battery depletion. The energy consumption highly depends on the protocols operating on the node. There is a need to improve the reliability and efficiency of the WSNs protocols in order to operate in a dynamic and changing environment.

### 1.1.1 Wireless Sensor Networks

Sensor nodes are originally designed to monitor and collect information from the environment and send the information to a central location over a wired network. These nodes usually rely on their on-board local power sources such as batteries or energy harvesting device for data collection, processing and communication. With the recent advancement in microchip and communication technology, the wired communication module has been replaced with a tiny radio transmitter that has the capability to transmit over the air (Chong and Kumar, 2003). Consequently, each sensor node becomes an independent and autonomous unit that has the ability to forward its data using the radio as a means of communication. This provides an extra level of flexibility for the sensor nodes to be deployed anywhere, anytime without the needs for expensive wiring. As a result, WSNs can be applied to various applications ranging from toys to household applications, from hobbies to safety critical applications.

### 1.1.2 Dependable Wireless Sensor Networks

For the WSNs to be applicable to safety critical systems and acceptable to the users, the individual nodes, including the hardware components and software running on it, must be made dependable to provide the level of service required for the application to function. According to Avizienis and Laprie (1986), a network can only be dependable if *"reliance can justifiably be placed on the services it delivers"*. This definition highlights the need for a network to have confidence and be trustworthy. A more measurable definition of dependability is stated by Avizienis et al. (2004a) as:

> *"The ability to avoid service failures that are more frequent and more severe than is acceptable to the user(s)."*

There is a need to avoid failure that may lead to negative consequences on the systems affecting their operations and dependability. Avizienis et al. (2004a) has described dependability as an integrated concept that constitutes of the following attributes namely availability, reliability, safety, integrity, and maintainability. In this thesis, we focus mainly on the operation aspects of dependability namely **Reliability** and **Availability**. Based on these properties, a definition of dependable WSNs for our research works is:

> *The ability for the networks to deliver the level of service, in term of packet reliability and network availability, specified by the application at a desirable level of confidence.*

### 1.1.3 Effect of Radio Abnormality on the Dependability of Wireless Sensor Networks

With the use of wireless channels, the radio and the packet data can easily be exposed and subjected to anomalies (Karlof and Wagner, 2003). According to Chandola et al. (2009), anomalies are *observations that do not correspond to a well defined notion of normal behaviours*. Anomalies in WSNs can be caused by errors, malfunction and attacks. These anomalies may cause the network software to function properly or even lead to network outage, affecting the dependability of the WSNs. The fact that the nodes share the same radio frequency spectrum used by other wireless devices cannot always guarantee a reliable communication, making them vulnerable to signal distortion, ambient noise and interference. Studies by Boers et al. (2010) have shown that interferences created by these wireless devices can disrupt the normal routing operation of the sensor node leading to significant packet losses and delays. As a reliable communication is an important aspect in the safety critical systems, a dependable routing is required that is resistant to the network anomalies related to radio interference.

### 1.1.4 Dependable Routing

To achieve sufficient level of dependability in WSNs, different fault prevention and recovery solutions have been proposed in the WSN literature to tolerate failure either through the development of new protocols or enhancement of existing protocols. Most of these works have focused on reactive routing due to its energy saving on-demand ability to discover a new route when communications between two nodes are required.

Reactive routing protocols such as Adhoc On-demand Distance Vector (AODV) have the ability to discover the route when required (Perkins and Royer, 1999). AODV uses the flooding mechanism to broadcast the route request to determine for new route during failure and can be very expensive to perform. Hence, it must be used sparingly. AODV does not distinguish failure. It relies on the link layer feedback and the distance traverse by the packet to determine whether to broadcast for new route or drop the packet. As WSNs are prone to different failures with different durations caused by neighbouring nodes, external radio devices, moving object and operating environments, nodes can suffer from transient, intermittent and permanent failure. A combination of these failures can occur and produce a complex unpredictable behaviour that cannot be addressed with a single protocol. For example, transient failures may trigger the link layer to notify failure to the AODV and result in route discovery. When the next-hop neighbour

experiencing the transient failure recovers, it will respond to the request while other nodes propagate the route request to all its local nodes. This will create a ripple effect that may congest the network. It is necessary to provide a reliable mechanism for the nodes to change their routing strategy according to the current network topology in order to re-establish the network connection. This leads to a motivation to investigate the potential of integrating different routing protocols to switch between different protocols for different interferences in order to answer the first research question below:

> *Research Question 1: Can we improve the dependability of WSNs by integrating and switching between different routing protocols using a multi modal approach to function according to its operating environments?*

### 1.1.5 Fault Tolerance Approach toward Dependability

In order to provide an appropriate routing strategy, the node must be able to identify and diagnose the failure quickly and reliably. It is necessary to build a fault tolerance network that has the ability to deliver the required service in the presence of faults (Avizienis et al., 2004a). The first step toward tolerance is detection.

Anomaly detection in WSNs has attracted significant attention over the last 10 years based on comparing the current network condition with a normal network pattern (Xie et al., 2011). Any deviation from the normal pattern will be declared as anomaly. To design an Anomaly Detection System (ADS) for WSNs is a challenging task due to the limited resources in the nodes. Most of the techniques investigated by Xie et al. (2011) are based on the assumption that the normal model does not change during operational time and it is generated offline. This is not practical in WSNs due to dynamic operating environments of the sensor nodes triggering a change in the normal model. The inaccessibility of the nodes and the limited network communication bandwidth can make the nodes difficult to be updated. Hence, the nodes should be able to detect changes in their environment online and update the normal model individually. Another problem with current WSN's ADS is that none of the fault tolerance detections has been implemented as most existing WSNs ADS highlighted by Karlof and Wagner (2003) are tailored toward malicious attacks.

Alternative fault tolerance approaches based on immune systems called Artificial Immune System (AIS) have been applied in a wide variety of application areas to maintain the systems integrity and functionality (Hart and Timmis, 2008). De Castro and Timmis (2002) define AIS as "*Adaptive systems, inspired by theoretical*

*immunology and observed immune functions, principles and models, which are applied to problem solving*". Over the years, many immune-inspired ADSs have been successfully applied to WSNs to detect and recovery from anomalies (Davoudani et al., 2007; Wallenta et al., 2010). This is partly motivated by the analogy between the characteristics of WSNs and the immune system such as distributed, autonomous, self-organise, scalable and prone to failures. As WSNs nodes are susceptible to interference from other radio emitting devices, there is a need to investigate the immune theories to integrate detection, diagnostic and recovery to rectify these failures and improve the response rate, energy efficiency and dependability of the WSNs. This motivates the application of AIS to address the second research question:

> *Research Question 2: Can an immune-inspired algorithm be applied to assist the routing protocol to detect and classify the different characteristics of the non-intentional interference in order to recover from network failures?*

### 1.1.6 Establishing Dependability with Confidence

The task to demonstrate dependable WSNs does not only involve the development of a reliable protocol, but also the evaluation process must provide a sufficient level of confidence that the protocol is dependable. Due to the high cost in deploying real sensor motes, researchers have used experimental tools such as simulator (NS2, 2002) and pilot study (Szewczyk et al., 2004) to evaluate the dependability of WSNs. Like any other software, the routing protocol must be tested comprehensively using those tools before it is deployed to remove any error that may be introduced during design and testing. There is a need to assess the confidence regarding the achievable dependability level of the routing protocols.

Designing a good testing strategy based on the real world is not always easy (Langendoen et al., 2006). WSN systems too often fail to provide expected results once deployed. It is complicated and complex to capture and model the real environment in simulation. Real Deployment usually fails even though it has been tested repeatedly. Langendoen et al. (2006) claim that the design and testing of WSNs protocols are hardened by the lack of effective testing and evaluation approach to produce a confident feedback on a critical figure of merit such as packet delivery reliability and network availability, that is necessary when trying to assess the network ability to perform within a persistent level of dependability across different operating environments. There is a need to conduct a comprehensive evaluation to analyse the level of tolerance on uncertainty that presence

in an experiment. Hence, statistical analysis is necessary to reduce the impact of uncertainty in order to achieve a sufficient level of confidence that is a representative of the prognostic results. A combination of simulation and sensor hardwares is used to evaluate the dependable routing protocol of WSNs to investigate the third research question:

> *Research Question 3: Can the dependability of network protocol be demonstrated by reducing the experimental uncertainty using state of the art statistical techniques?*

## 1.2 Statement of Hypothesis

Based on the motivations and research questions highlighted in the previous section, the hypothesis of this thesis is formalised as follows:

> *Dynamically switching routing protocols can improve a Wireless Sensor Network's performance based on immune-inspired monitoring, however understanding its performance requires a systematic statistical evaluation.*

## 1.3 Contributions

In Chapter 2, a survey on the existing literature has revealed that the components of the WSNs are unreliable and prone to failure. The issues of dependability have been widely ignored. Failures can occur in different parts of the WSNs. A causal link between the failures is identified. In order to satisfy the dependability of the emerging sensor network application, the key design properties required for a dependable WSN are proposed namely: fault tolerance, energy efficiency, scalability and adaptability. To evaluate the fault tolerance approaches applied in WSNs, a comprehensive survey is conducted. The results from the analysis have shown that there are still some gaps in the existing fault tolerance techniques applied in the WSNs protocols such as the ability to adapt to different failures, integrated fault tolerance to repair the network, and dynamic detection model.

It is necessary to construct an experimental testbed and evaluate the dependability of the WSNs protocols to identify the limitation highlighted in Chapter 2. A dependability assessment of existing protocols in WSNs is presented in Chapter 3. The methodology employed to evaluate the performance of the WSNs is described. An interference model based on the ON/OFF model to simulate the failures generated by radio emitting devices is proposed. The results from the

simulation on different routing protocols have demonstrated that a routing protocol does not always give the same performance when different failures are introduced. One protocol may outperform the other in condition X, but the opposite in Condition Y where X and Y represent different failure conditions. There is a need to adapt the routing protocol according to the failures.

The main contribution of the work is presented in Chapter 4 to address the single modal issue highlighted in Chapter 3. A multimodal technique to switch between different operating modes is proposed in order to adapt the changing environment. This approach is implemented in the network layers using the reactive routing protocol as a case study. A Multi-modal Routing Protocol (MRP) is presented and evaluated on a simulator to measure its reliability and performance. The simulated results have shown that the multimodal approach has a higher dependability compared to a single mode.

The second main contribution in Chapter 5 applies the immune-inspired algorithms to assist the MRP to provide a co-ordinated automated recovery. An AIS algorithm, the Receptor Density Algorithms (RDA) proposed by Owens et al. (2012), is extended to identify the failures and interact with MRP to provide a response in order to rectify the failure. The Interference Detection and Recovery Systems (IDRS) is proposed and evaluated in real hardware and simulation. The IDRS has improved the response provided by the routing protocol.

Chapter 6 presents a systematic evaluation approach to demonstrate the dependability of the protocol using non-parametric statistical techniques. A Systematic Protocol Evaluation Technique (SPET) is proposed to reduce the uncertainties in the experiment and to assess the confidence of the measurements taken from the experiments. The Conceptual Statistical Test Framework (CSTF) is proposed to cross-validate the results obtained from both hardware and simulations in SPET. By using SPET, the dependability of the protocol can be evaluated and the confidence level on the dependability can be measured.

## 1.4 Organisation of Thesis

- Chapter 2 introduces the WSNs and the importance of dependability in WSNs. It presents the threats in WSNs that can affect the dependability of the network and investigates some of the fault tolerant approaches applied to WSNs. A survey of existing routing protocols and the current state of the art approaches to evaluate the performance of the routing protocols is presented.

- Chapter 3 provides some of the existing methods to evaluate the performance of the WSNs. A comparison between four commonly used reactive routing protocols is evaluated by simulation to assess the ability to rectify anomalies in a dynamic environment.

- Chapter 4 proposes the Multi-Modal approaches to improve the fault tolerance of the routing protocol in a dynamic changing environment. It re-uses and integrates existing reactive routing protocols presented in Chapter 3 to switch between different operating *modes* to handle different failure conditions. The robustness and scalability of the routing protocol are analysed and evaluated using a simulator.

- It is necessary to determine the cause of the failure online in order to provide a quick and effective response. Chapter 5 introduces an immune-inspired algorithm called the Receptor Density Algorithms (RDA) to classify the current radio patterns collected by the nodes to improve the recovery. The RDA is integrated with MRP to provide an automated self-healing system to detect, diagnose and respond based on the current state of the networks. The reliability and the overhead generated by IDRS are evaluated against MRP in both real hardwares and simulation.

- In order to demonstrate the dependability of the proposed MRP, we propose a statistical approach to evaluate the WSNs in Chapter 6. It applies the concept of Correctness, Consistency, Completeness, and Cost-effectiveness (4Cs) to measure the effectiveness of the different evaluation strategy. A systematic protocol Evaluation Technique (SPET) is proposed in order to establish confidence in both the simulation and the real world WSNs deployment. The MRP is used as a case study to evaluate the benefit of SPET.

- In Chapter 7, the thesis is concluded with a discussion of the main contributions and potential avenues for future work.

CHAPTER

# TWO

# Towards a Dependable Wireless Sensor Networks

In order to understand the issues of dependability in WSNs, it is essential to introduce the fundamental components in Section 2.1, and present the properties of the WSNs in Section 2.2. We highlight the needs of a dependable WSNs in Section 2.3. The concept of failure in WSNs is formalised in Section 2.4 before different failures in WSNs are investigated in Section 2.5. Section 2.6 highlights some of the existing fault tolerance approaches to detect, diagnose and recover from these faults and identifies some of the limitations of existing approaches. In order to investigate the potential of the application of immune-inspired solution to address the limitations, an introduction of the immunological theory and a review of existing algorithms applied in a fault tolerance system are presented in Section 2.7. As the routing protocol plays an important role in ensuring the operation of the WSNs, different reactive routing protocols are introduced and compared in Section 2.8. In order to evaluate the dependability of these routing protocols. existing state of the art experimental approaches is presented in Section 2.9. This chapter concludes with a summary of the key challenges in Section 2.10.

## 2.1 Wireless Sensor Networks

WSNs consist of spatially distributed autonomous wireless sensing devices that interact with the environment to capture and send the data to the user through

a computer system (Akyildiz and Vuran, 2010). It may range from tens to thousands of sensor nodes distributed across a wide area. The nodes are usually deployed in a predefined location to provide complete coverage (Zou and Chakrabarty, 2007). During operation, a source serves as an interface to the real world to monitor and collect environmental information. This information is processed into data packets before they are transmitted to the sink or base station via a single-hop or multihop network depending on the distance between the source and the sink as shown in Figure 2.1. Single-hop network provides a better data delivery reliability than multihop network as each transmitted packet can immediately be acknowledged by the sink upon receiving the data packet (Brownfield et al., 2006). However, the source may its transmission range if the destination is outside the boundary of the transmission range of the source, requiring more energy resources that may reduce the network life (Raghunathan et al., 2002). In contrast, a node in a multihop network can communicate with other nodes at a lower transmission power as long as the two nodes are within their radio transmission range. This reduces the energy consumption in the node. Hence, the multihop network is the preferable network that will be investigated in our research works.

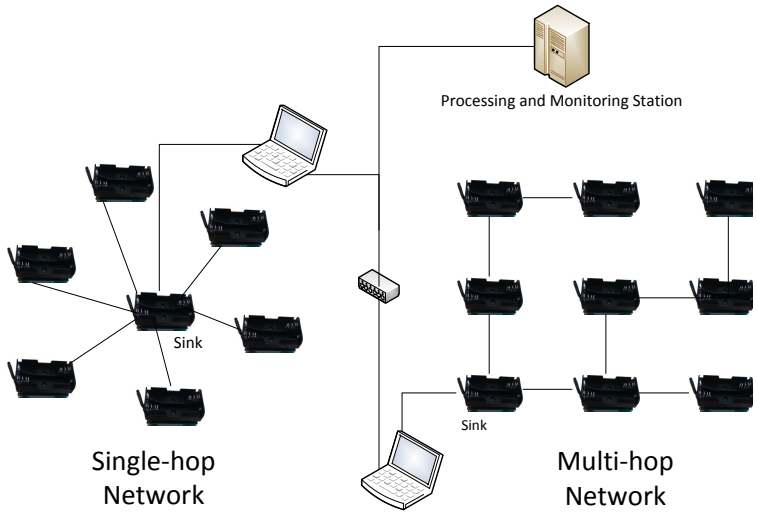
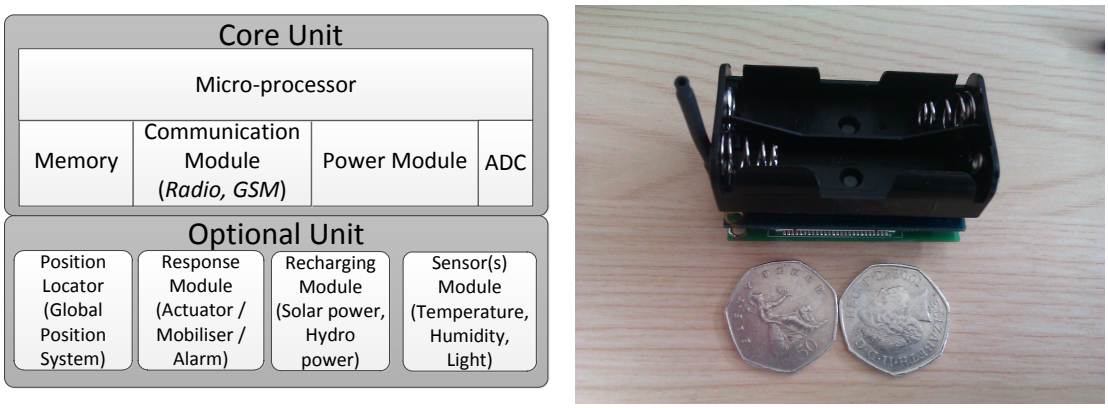



Figure 2.1: WSNs deployment showing the single-hop (left) and multihop network (right). In a multihop network, a routing protocol is required.

As shown in Figure 2.2(a), each mote can be divided into two units: the core unit and the optional unit. The core unit hosts the main components required for the mote to function. The optional unit may consist of sensors board, charging module, position locator and response module (Akyildiz et al., 2002). The microprocessor of a typical node shown in Figure 2.2(b) usually has a limited processing power to run the systems software and process the sensed data. The

memory module is usually small in size of around 10K bytes of RAM and 100K bytes of flash RAM for data, configurations and application storage. The communication module consists of a radio transceiver and is used to transmit or receive packets. The mote can either be powered directly from an energy source such as a computer or from a battery source (Sohraby et al., 2007). The lifespan of a battery powered node highly depends on the energy utilisation consumed by the operations of the individual node. Energy harvesting techniques such as solar and vibration have been proposed as an alternative energy source. However, Anthony et al. (2012) have shown that sensor motes cannot always benefit from these technologies as these energy sources are not always available for charging. As a result, an energy efficient protocol is necessary to prolong the operations.




(a) The components of a mote (b) A Typical commercial node (TelosB mote)

Figure 2.2: An example of WSNs Hardware

For a node to perform the task assigned, specialised system software are installed and run on the mote to regulate its operation. Due to the limited resources in the node, these software are usually designed with careful consideration on the resource availability with minimal complexity.

### Communication Stack

WSNs build on the IEEE 802.15.4 multi-layered standard for low-rate wireless personal area networks (WPAN) to provide the physical and link layer framework for low data rate communications systems and low power consumption application (Howitt and Gutierrez, 2003). Additional network and the transport layers are defined by the WSN industrial standards. The functionalities of each layer are described below.

- *Physical layer (PHY)* acts as an interface to radio channel and controls the functions related to the radio frequency (RF) transceiver such as the transmission and reception of the packets, the clear channel assessment (CCA),

and RF energy detection (ED). Simple data modulation, transmitting and receiving techniques have been used to reduce complexity and energy requirements of the WSN. CCA detects the channel occupancy by performing ED or Carrier Sense (CS). Commercial sensor node shares the same RF with other wireless devices and operates at 2.4 GHz, using the 16 channels available under the IEEE 802.15.4, each labelled sequentially from 11 to 26 (IEEE, 2006a). As illustrated in Figure 2.3, some of these channels overlap with other home devices like Wireless Local Area Network (WLAN) and can interfere with the radio and the operation of the nodes. Although industrial standards such as WirelessHART (Song et al., 2008) and ISA100.11a (ISA100, 2009) address the issue of interference by employing Frequency Hopping Spread Spectrum (FHSS) where the channel used for data transmission will alternate in random sequence at packet level, an agreement on the frequency to use is necessary. It does not overcome the interference from other devices using similar approaches such as Bluetooth. Hence, it is critical to identify and overcome these interferences in order to reduce failures. The frequency hopping mechanism can be applied for recovery.

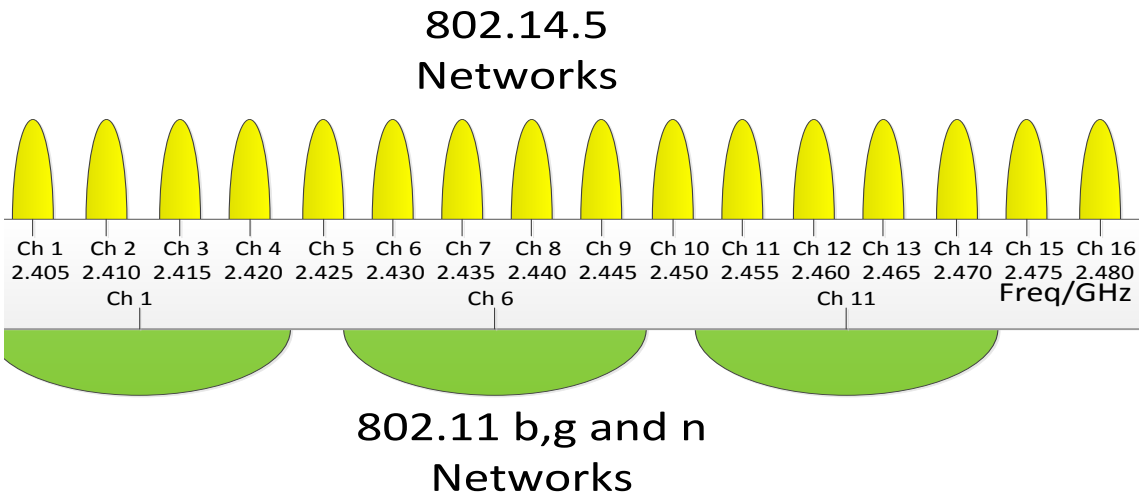



Figure 2.3: Frequency Allocation for WSNs starts from channel 11 to 26. Channel 26 is usually used in WSN experiment as it is not utilised by WLAN (802.11).

- *Data link layer* provides functions such as Medium Access Control (MAC), error detection and control. It is responsible for the establishment of a reliable link between the nodes. The MAC protocol plays an important role in deciding the network protocol to be implemented in the layer above in order for the communication stack to function correctly. The topology of the network is also dictated by the MAC protocols used in which protocols

such as the Time division multiple access (TDMA) usually has hierarchical topology while Carrier sense multiple access with collision avoidance (CSMA/CA) is mostly flat-based. The IEEE 802.15.4 uses the CSMA/CA as the default MAC protocol. It is flexible and does not require any time synchronisation (IEEE, 2006a). Assigning an effective time schedule in TDMA can be very difficult and expensive to perform in WSNs especially in a large network. Hence, the CSMA/CA protocol is used in this thesis.

- *Network layer* hosts the routing function including the network addressing, packet forwarding and the error recovery. It is dependent on the application requirement and network architecture. The routing function usually relies on the link layer to detect failures. It maintains the quality of service required by the transport and application layers. Although conventional data distribution approaches like flooding and gossiping can be applied to send the data across the network, it can create unnecessary wastage in the communication bandwidth and energy by sending redundant information across the networks (Zhang et al., 2009). Hence, a routing protocol is necessary to forward the data packet across the multihop network, control network flow and reduce communication overhead.

- *Transport layer* maintains the reliability and quality of data flow between nodes. It provides packet-loss recovery and congestion control mechanism caused by network congestion, packet collision, buffer overflow and node failure. Currently, the two approaches applied are based on the node-to-node or end-to-end reliability. In the node-to-node reliability, intermediate nodes in the networks store the forwarding packet in their buffer (Wan et al., 2005). The end-to-end reliability uses an end-to-end transmission in which the source will keep all the transmitted packets until an acknowledgement is received (Iyer et al., 2005). Both protocols require communications between the sender and receiver to update the delivery status. Due to the limited memory space to buffer the forwarding packets, industrial standards such as like ZigBee and Specknet have omitted the transport layer (Arvind and Wong, 2004). They attempt to provide the reliability function at the lower layer of the communication stack to reduce the processing and communication overhead between layers.

**Applications layer**

The application layer hosts the software that is responsible for presenting the information to the users, provides interface to the environment and accepts re-

quests from the layer below. Langendoen (2008) classifies the types of WSNs applications into periodic-based and event-based. In periodic-based application, data is captured and sent to the end users periodically, for instance every 30 seconds for a pulse oximetry application (Chipara et al., 2010). In event-based application, the source only detects and sends critical information to the users when an unusual event is detected by the sensor (Liu et al., 2010). In a static network, the periodic-based application usually discovers the path to the sink when it is first set up and is only updated when there is a communication failure along the route. In contrast, the routing information for an event-based application may need to be checked or updated more frequently to ensure that the path to the sink is still exist and the routers are alive. Both types of applications generally require to operate over a period time in order to provide the services required.

## 2.2 Design Challenges in WSNs

In recent years, a large body of research in WSNs has evolved in improving the communication protocol in order to prolong the operation of the networks (Hatler, 2012). Only a limited number focuses on improving the reliability and robustness of the WSNs. During the development of any WSNs protocol, it is essential to address and incorporate the following properties of the WSNs in the design:

P1: The *resources availability* such as memory and processing, communication bandwidth and the radio capability on the nodes. The nodes are usually *resource constraints*.

P2: The *high density* of the sensor nodes.

P3: The *uncontrollable operation* of the wireless medium due to interference, fading and contention making the network *prone to failure*,

P4: The *dynamic topology* of the network that is *distributed* and *autonomous* without a centralised controller.

P5: The *realisation* of the reliable operation during real deployment network.

In order to satisfy the dependability of the emerging sensor network application, the design of the WSNs hardware and system software must exhibit the following properties including fault tolerant, energy efficiency, scalability and adaptability.

### 2.2.1 Tolerate failure

In a fault-tolerant network, the nodes must have the ability to deal with nodes or links failure to maintain reliability and provide continuous operation of the networks. Explicit fault tolerant can be provided if there are a sufficient number of redundant nodes available to provide the coverage required for sensing or communication (Luo et al., 2006). However, to determine the minimum number of the nodes without the nodes interfering with each other can be a challenge. As a result, implicit detection and recovery are necessary to improve the reliability and robustness of the nodes to failure. One technique to achieve reliability is by retransmission (Balakrishnan et al., 1997). Zheng et al. (2011) highlight that retransmission in WSNs can improve the packet delivery when wireless channel is in a mild or bad condition. However, maximising the reliability through retransmission may increase the network energy consumption substantially and increase packet delay that may affect the timeliness of the packet received (Fonseca et al., 2006). Depending on the type of applications, timing delay may dictate the performance and stability of a control system (Moyne and Tilbury, 2007). Hence, the network designers need to consider the trade-off between reliability and energy consumption as higher reliability may result in higher communication overhead and hence energy consumptions (Zheng et al., 2011).

### 2.2.2 Efficiency

As the nodes have limited processing power, small memory storage, shared transmission bandwidth and are usually battery powered, each node must manage and utilise these resources effectively. Consequently, protocols used in WSNs have to be lightweight and energy-efficient to prolong the availability of the node to provide the services required by the application. As the requirement attributes for the safety critical application such as high reliability and low latency may require significant energy, these attributes must be prescribed accordingly to meet the level of assurance required for the application to meet its target. These applications can usually tolerate up to a certain degree of packet losses (10-20%) and delays (Schenato et al., 2007; Moyne and Tilbury, 2007). Hence, maximising packet delivery while minimising the delay to the optimum level is not always necessary since these strategies can significantly reduce the network lifetime. As the radio communication is a major energy consumer in WSNs, it needs to be optimised. Most existing literature has focused on energy-efficient networking protocols for WSNs. According to Sohraby et al. (2007), energy efficiency in WSNs can be achieved in three ways:

- Duty cycling with periodic switching between the node's operation modes in order to conserve energy and increase the life time of the network.
- Local processing to reduce the energy required by radio transmission.
- The use of multihop network to minimise the long-range transmission range. However, Tate and Bate (2010) highlight that multihop generates more messages that may cause more collision and hence higher energy consumption.

### 2.2.3 Scalability

Scalability is another challenge due to the large-scale and dense deployments. WSNs can be deployed indoor or outdoor in large scale in the order of hundreds or thousands. The variation of the number of nodes in the network highly depends on the application running on it. The nodes are usually deployed in close proximity to each other to achieve a higher sensing coverage and redundancy (Chipara et al., 2010). Hence, the protocols should be able to adapt to variation in the network size and avoid inter-nodes interference.

### 2.2.4 Adaptability

Adaptability is the other challenge due to the lack of infrastructure and network dynamics. WSNs do not have a fixed infrastructure as the network logical topology can change due to the unreliable communication links or nodes that fluctuate between connecting nodes. Every node requires to self-organise and operate independently in order to be flexible to the continuously changing environment. The system software in the node need to respond to both internal stimuli generated within the node and external stimuli generated from the environment. The internal stimuli may be generated due to the changes in application's requirement or an error in the system software, while the external stimuli is produced due to the time varying wireless channels, the dynamic variations of the network or the occurrence of events. Hence, each node must possess self-adaptive and self-healing capabilities to cope and adapt to spatial and temporal changes of the operating environment.

### 2.2.5 Discussion

As a result of these properties, the design of the WSNs systems components has to take into accounts a number of development factors such as the correct implementation, operation and fault tolerance of the system to provide dependable

service. As the communication protocol forms a major part of the WSNs, it is critical to design and evaluate the effectiveness of the communication protocol that satisfies the application requirements and optimises the energy efficiency of the systems. Safety Critical Systems (SCS) such as the fire detection and rescue and the health monitoring usually consist of a set of measurable service attributes imposed by the applications in terms of reliability, availability, delay, and energy consumption. These attributes can be applied to assess the dependability of the WSNs by ensuring that these requirements have been met at a sufficient level.

## 2.3 Dependable WSN

To allow for the application of WSNs into SCS domain, a high degree of dependability is necessary. These safety critical applications demand not only energy efficient operation to prolong operation, but also strict data delivery performance. In particular, data must be transported to a sink in a timely and reliable fashion. Based on the definition of *dependability* given in Section 1.1.2 of Chapter 1, it can be comprehended as the ability for the networks to provide the packet reliability and network availability specified by the application at a desirable level of confidence despite of the occurrence of faults in the networks. The results from previous studies have shown that there is growing evidence where the deployment of WSNs in real world is unreliable. Some of the failures highlighted include packets loss, communication failures induced by the operating environments, and short term availability caused by power exhaustion (Tolle et al., 2005; Huo et al., 2009). Despite the existing success stories of achieving dependability for the traditional SCS, the solutions proposed are not easily transferable to the resource constraint WSNs. The classical approaches often require complex software that cannot be performed online in WSNs. As a result, alternative dependability enhancing techniques such as fault detection, identification and recovery are essential to maintain or increase the acceptable level of dependability in WSNs.

## 2.4 Failures in WSNs

Before fault tolerance techniques can be applied in WSN, it is necessary to understand the failure characteristics of WSNs. The WSNs are found to be unreliable in existing literature (Langendoen et al., 2006; Tolle et al., 2005). The application and communication protocols running are primarily designed to be energy efficient using non-complex algorithms in order to reduce the energy consumption and computational cost (Langendoen et al., 2006). The hardware may be deployed to

operate in an unattended manner for a long period of time and exposed to different environmental condition that may deteriorate the node's hardware components (Tolle et al., 2005). For these reasons, WSNs are prone to failure.

### 2.4.1 Fault, Error and Failure

According to Avizienis and Laprie (1986), a fault is a defect or incorrect state of the hardware or software that can occur in any part of the system. This fault is usually untraceable until an error has occurred. An error corresponds to an incorrect and unknown state of the system that can lead to failure. A failure may occur if the error is not detected and rectified (Laprie, 1985). When a failure occurs, the system may deviate from its specification and unable to deliver its intended functionality. An unattended failure may later lead to further faults and errors of the system forming a chain reaction as depicted in Figure 2.4. An error is the manifestation of a fault in the system, and a failure is the result of such error on the service (Avizienis and Laprie, 1986).

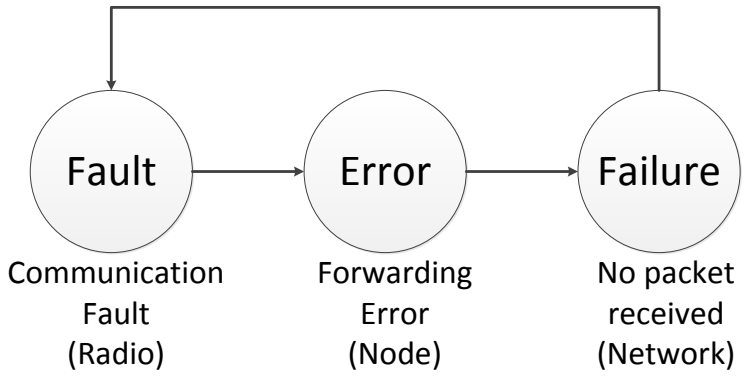



Figure 2.4: Formation and manifestation mechanisms of faults, errors, and failures: The arrows show the relationship between the three states and the fields in the brackets show the location where the abnormalities are experienced.

For example in Figure 2.5, an application running on node 1 is required to send sensor readings periodically to node 4 via node 2 and 3. As node 2 is about to forward a data packet, the node's radio channel in node 3 may experience a radio communication fault induced by external interference (*fault state*). Since node 2 does not detect the presence of interference in node 3, node 2 will attempt to send the packet that may result in irregular packet loss (*error state*) due to the stochasticity of the interference. As a result, the application running on node 4 will not be able to receive the data packet and an irregular network outage (*failure state*) is experience by the users. This erroneous condition may lead to further application fault if the degrading network condition is not detected and rectified.

Therefore, it is critical to detect these irregular patterns and recover from the *fault* at the earlier stage to prevent application failure.

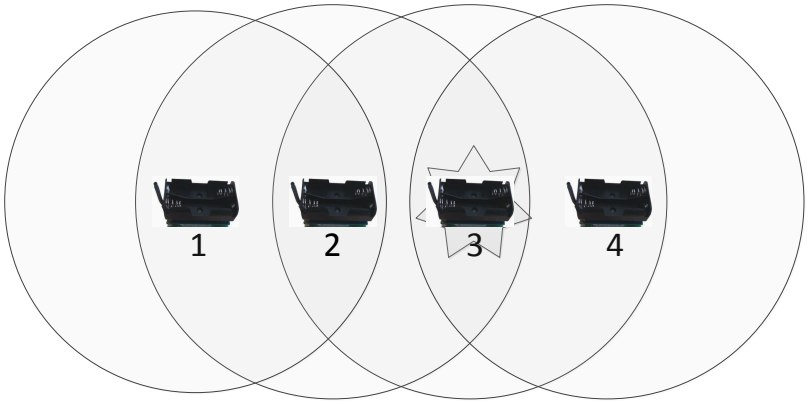



Figure 2.5: Examples of manifestation of faults caused by communication fault in WSNs: Node 1 is periodically sending packet to node 4 via node 2 and 3. Node 3 experiences radio fault that lead to packet forwarding error in node 2 and eventually fail network and application if the error is not rectified.

### 2.4.2 Fault Characteristics

Siewiorek (1990) classifies faults into three types:

- *Permanent*: Fault, usually in hardware, that is continuous to appear until the faulty component is replaced. It is considerably easier to detect.
- *Intermittent*: Fault that occurs occasionally due to an unstable system state.
- *Transient*: Fault that occurs temporarily due to environmental condition.

Based on the given definitions of faults, it is difficult to distinguish the temporal characteristics of the intermittent and transient faults and provide a prognosis of these faults. The ability to detect a transient fault that can lead to reduced performance and eventual failure of the network is vital as the response required to rectify the fault can be made before it fails the network. Siewiorek (1990) makes the distinction based on the ability to detect the fault. An intermittent fault is usually detectable as it exhibits a repeatable failure pattern but a transient failure is more difficult to detect and recover. For example, the intermittent fault can be discriminated by evaluating the rate of fault occurring which is relatively high (Bondavalli et al., 2000). In contrast, transient failure is usually related to temporary environmental factor that is usually rectified by the system software or when the source of error disappeared. However, due to the design of the system

software, this error may trigger inappropriate response that may aggravate the networks (Tseng et al., 2002). For example, the self-reactive nature of the routing protocol to cope with transient error (obstruction) has resulted in congestion and network outage (Szewczyk et al., 2004). It can be very difficult to detect and predict the outcome of the transient fault as it adverse effects are usually irregular and may disappear rapidly (Bondavalli et al., 2000). However, the reaction of events generated by the fault on the network is conspicuous. For instance, faulty radio communication might not be directly observable but the unusual low number of packet errors in the forwarding node of the WSNs (i.e. the data delivery) can be observed. Bondavalli et al. (2000) highlight that the detecting a fault may require repetitive testing at the discrete time interval. By analysing the changes in data flow within a time window, it is possible to detect that an error has occurred.

Due to the complex and dynamic characteristics of WSNs, it is important to detect and determine the irregular transient fault at run-time to prevent network failure. It is crucial to understand the type of faults that can occur in WSNs to ensure that the right technique can be apply to tolerate them. These faults, also known as anomalies, are discussed in the next section.

## 2.5 Type of Anomalies

WSNs are commonly deployed in a harsh environment where each node can be subject to anomalies. These anomalies in WSNs can occur in the hardware as well as in any of the communication protocols. Physical hardware faults are usually induced internally that may affect its components and system software due to unreliable hardware and limited resources (Langendoen et al., 2006). The interaction faults are errors that are generated externally affecting the communication protocols, for instance, radio interference in the channel affecting the links. In order to address the research questions in Chapter 1, it is crucial to understand these anomalies in the WSNs domains to identify the failures that may degrade the WSNs protocol. From literature survey conducted on the previous deployments of WSNs, anomalies can present in the hardware, software, network, communication, or data. The following subsections investigate the four types of anomalies that have occurred in the real deployment of WSNs as a result of the faults.

### 2.5.1 Data anomalies

Data anomalies are errors that occurred in the information received in the sink. Inconsistency in measurement is usually considered as anomaly and needs to

be detected and eliminated. Data anomalies can be caused by processing faults (Elnahrawy and Nath, 2003), hardware fault (Ni et al., 2009) or malicious attack (Wang et al., 2006). Elnahrawy and Nath (2003) have classified the sources of data errors as systematic and random errors. Systematic errors are caused by changes in the operating conditions and calibration errors of the sensor node. Random errors are usually introduced by external noise such as environmental interference and inaccuracy to measurement and computational techniques. These errors can affect the accuracy of the sensor reading. For example, Tolle et al. (2005) reported that the data obtained from the sensors were out of normal range and did not correlate with the each other when the voltage of the battery fell below 2.4 volts. Systematic errors may be generated by long operation of the nodes that may affect the component or applications running on the nodes, such as the calibration of the sensors or timing drift in synchronisation (Ni et al., 2009). Data anomalies that are produced intentionally can occur when an attacker injects false packet or retransmits packets into the network after gaining access to the sensor node (Wang et al., 2006). This attack can deplete the battery of the node and cause network congestion (Raymond and Midkiff, 2008). However, intentional data anomalies in real have not been reported and are only performed in test lab or simulation to show the ability of WSN to avoid threat against malicious attacks.

### 2.5.2 Node anomalies

Node anomalies occur when the node is unable to perform the tasks it is assigned to. Node anomalies can be caused by hardware (Szewczyk et al., 2004) or software faults (Werner-Allen et al., 2006). The node is usually manufactured at low cost and can be easily damaged due to impact or extreme weather condition. During deployment, sensors may be dropped from high altitudes damaging the outer protective enclosure and exposing the internal components of the sensor nodes (Kahn et al., 1999). Sensors may be damaged or disconnected from the node due to ground impact causing intermittent or continuous false alarms or misses. For instance, node failure due to short circuit has been reported by Szewczyk et al. (2004) when the circuit board of the sensor node was exposed to direct contact with water. The low quality hardware and appropriate antenna design are also reported by Langendoen et al. (2006) as the contributing factors to the node failure for a large scale deployment of potato cultivation where no useful data were captured. Some nodes may run out of power due to the limited on-board energy resources (Tolle et al., 2005). The limited computational and storage resources available on a node may restrict the amount of processing that can be performed

at the node. If this limit is exceeded, the tasks may not complete causing non-deterministic behaviour and segmentation fault in the software. As the nodes usually operate in an inaccessible location, maintenance steps, software updates and bugs fixing are difficult and not usually performed.

### 2.5.3 Network anomalies

Network anomalies can be caused by node failure, gateway failure, unavailability of transmission radio, or malicious attacks. Most network anomalies found in the literature are attack-related as existing routing protocols and isolated deployment of the nodes can be easily exploited (Karlof and Wagner, 2003). Routing is one of the essential building blocks for a multi-hop WSNs. The broadcast nature of the routing protocol can result in packet losses from collisions when two nodes within range of each other transmit simultaneously. Collided packets can be lost at a receiver due to the hidden terminal effect (Woo and Culler, 2001). This must be detected by the sender for retransmission and to ensure that the data packet is forwarded and send to the sink. In multi-hop WSNs, the radio communications and the network connection is highly volatile and cannot always guarantee the same packet delivery rate in real deployment as in the test lab (Szewczyk et al., 2004). Ingelrest et al. (2010) have highlighted that the unstable link between nodes can cause constant changes in the routing table. Constant routing table update may require additional radio communication, occupying the shared radio channel and consuming additional energy resource in the nodes. Hence, an intelligent method to determine the cause of the unstable link is necessary to ensure that route update is only performed when necessary, for example when the link degradation has a significant effect on the reliability or the failure is permanent.

### 2.5.4 Signal anomalies

Wireless medium can be greatly affected by noisy environments causing the signal to attenuate or loss. Radio irregularity and channel interference are common phenomena in wireless transmission that may affect the radio operation of the node. Zhou et al. (2004) highlight that messages may be lost as a result of fading during propagation over the wireless medium. These irregularity and interferences are created by the non-isotropic characteristics of the propagation media and the heterogeneous properties of the nodes that may differ according to the radio module. The communication can also be interrupted by physical phenomena, metallic object, other radio emitting devices and hardware calibration error

such as antenna gain, and fluctuation of the radio transmission range due to batteries depletion (Tolle et al., 2005). An object that is larger than the wavelength of the signal, such as a building or tree, can reflect the signal propagating through the wireless medium causing interference in the network. Such fault is always transient in nature as these interference are non deterministic as their presences fluctuate (Chipara et al., 2010). In a potato field, the radio transmission range between two node can be obstructed by the plants as each plant grows, affecting the link quality as observed by Thelen et al. (2005). Equipment and machinery operating at the same frequency with the WSNs node can generate noise in the network causing transient faults (Gungor and Hancke, 2009). WSNs also operate in unprotected wireless communication channel that is shared by other wireless devices such as microwave and wireless network, making the node vulnerable to interference. These interferences can lead to packet collision and increase the packet error rate (Gungor and Hancke, 2009; Huo et al., 2009). These interferences must be detected and avoided to ensure reliable communication.

### 2.5.5 Discussion

From the reviews, anomalies can occur in any part of the WSNs including the application's functionality and data, the node, the networks and transmission channel. There is some causality between these anomalies where the occurrence of these faults may have a knock-on effect on each other. Using the communication stack model described in Section 2.1, the causality link between the anomalies can be shown based on the layered approach as exhibited in Figure 2.6. The top layer anomalies may occur as a consequence of the faults generated by the lower layer. For example, due to the anomalies produced in the radio communication signal, the routing protocol in the network may not be able to function correctly. As a result, the node cannot forward the message to the sink causing the application unable to detect and response to the event detected by the sensor. Hence, it is important to identify and recovery from these errors at the lower layer of the communication stack.

## 2.6 Achieving Dependable Network using Fault Tolerant Approaches

In a distributed system such as WSN, failure in one component may trigger a partial failure within the systems (Tanenbaum and Van Steen, 2002). This partial failure may or may not affect the operations of the other components depending

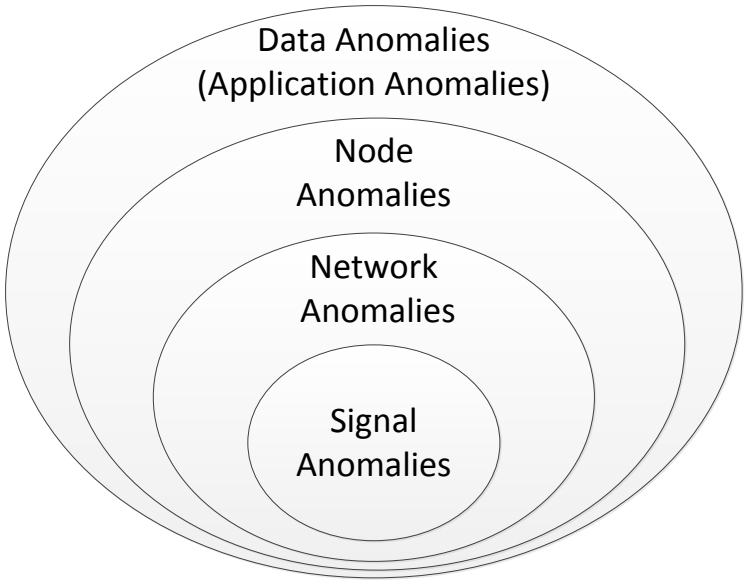



Figure 2.6: The causal link between the anomalies: The occurrence of the anomalies in the inner layer may cause anomalies to occur in the outer layer.

on the nature of the faults (Avizienis et al., 2004a). Tanenbaum and Van Steen (2002) state that it is critical to construct a system that can automatically detect and recover from partial failures without affecting the overall performance. For instance in WSNs, the application running on the node should continue to function in an acceptable level while the networks are being repaired. Each node should have the ability to tolerate failures and can continue to operate in a reliable manner even in the presence of faults. Hence, the reliability and availability of services provided by WSNs highly rely on the fault-tolerant ability of the node as it cannot be assumed that all sources of error can be eliminated after thorough engineering and intensive evaluation process. Each node must determine and remove the error by using error processing technique in order to provide a continuous and reliability service, (Laprie, 1985). The error processing technique consists of a set of actions namely:

- Error detection: the ability to detect the fault,
- Error diagnosis: the ability to identify the fault,
- Error recovery: the ability to rectify the fault.

These three actions are part of the fault tolerance approaches proposed by Laprie (1985) to achieve dependability. According to Avizienis et al. (2004a), the fault tolerance approaches can prevent network outages in the presence of faults. It attempts to address the issue of delivering a reliable and trustworthy service

during operation at a degraded level acceptable to the users. We describe some of the fault tolerance approaches applied in the WSNs context in the next few subsections.

### 2.6.1 Fault tolerant architecture

There are three architectures for deploying an agent to perform the fault tolerance namely centralised, distributed and hybrid architecture.

- ***Centralised approach:*** In centralised architectures, the anomaly detection is performed at the sink. WSNs collect information from the sensor nodes and send this information to the sink to be processed and analysed. Gupta et al. (2007) utilise this information to generate the global view of the network to detect and identify the anomalies. The sink usually has more resource available to perform complex detection algorithms to improve the detection accuracy. The sink also has the capacity to store historical data that can assist in detecting the anomalies. However, high volumes of data transfers are required to send these data (Ngai et al., 2006). A node cannot afford to periodically send its state and data as the radio communication usually consumes more energy than local processing (Raghunathan et al., 2002). The centralised approach also suffers from a single point of failure. The nodes within a region will fail quicker due to a higher concentration of data and management traffics. Rajasegarar et al. (2008) address this issue by aggregating the data within a local network to reduce the communication overhead. Although data aggregation can reduce the packet size and the number of the packets in the networks, it may remove critical information necessary to determine the causes of anomalies. The centralised approach may also increase the detection time and the Mean-Time-To-Recovery (MTTR) if the sink is far away from the anomalous node. MTTR represents the average time required to repair a failure which can be critical in bringing the network back to operation. This approach becomes inefficient and communication expensive as the network size increases.

- ***Distributed approach:*** In distributed architectures, the detection agent is installed in every node and promotes the concept of local decision-making. A node can monitor and obtain the local state of the neighbouring nodes within its transmission range to detect any abnormal activities. It allows a local node to make recovery decision. This decision making and recovery process is usually transparent to the sink. The communication between the

sink is only needed if the error cannot be resolved locally and a global recovery is required (Rajendran et al., 2004). This approach can reduce the communication overhead and improve the energy efficiency. To perform real time anomaly detection, da Silva et al. (2005) apply a distributed rule-based detection model to perform data anomalies detection. Each node generates the normal data pattern using its own data and the data collected from its neighbouring nodes. During operation, the current data pattern is analysed to detect any deviation from the normal pattern. Once anomalies have been detected, an alarm is sends to the sink or neighbours for consensus. This method is also applied to detect anomalous network behaviours (da Silva et al., 2005; Onat and Miri, 2005a,b). The distributed approaches can provide online detection but may require the node to store historical information to perform the anomaly detection. The amount of data to be stored highly depends on the algorithm applied and is restricted to the storage space available. Multiple alerts may be triggered during detection resulting in a sudden burst of traffics and packets collision. Additional energy consumption may be needed to listen to the network to collect the data periodically or continuously. As a node has limited resources to perform these tasks, a lightweight algorithm is usually applied.

- ***Hybrid approach:*** In hybrid architectures, the detection is performed locally and the recovery action is triggered by the sink or Cluster Head (CH). The WSNs can be grouped into smaller networks where a node within a group is assigned as detection agent. Hai et al. (2007) cluster the nearby nodes in the networks into subgroup. The sink randomly selects a node to become the CH. The CH monitors the nodes within its cluster and sends an alert to the sink when an anomalous behaviour is detected. Upon receiving the alert, the CH or sink can response to the failure. This approach can reduce the communication overhead but create additional processing overhead in the CH to carry out detection. Load balancing approaches such as round robin are usually applied to share the role of CH and distribute the processing evenly (Nam and Min, 2007). Hai et al. (2010) have shown a significant reduction in energy consumption in the node compared to a distributed approach when all the nodes are periodically activated as the detection agent based on the coverage area. However, this architecture suffers from a single point of failure as the detection is performed centrally in a localised distributed manner by the CH.

***Discussion***

The comparison between the three different architectures is provided in Table 2.1. Distributed approach is more efficient compared to centralised approach as less communication is required. It is important to minimise the communication overhead between nodes to extend the life of the network. To minimise these overheads, a distributed approach with load balancing is usually taken by ADS as the processing overhead requires to perform the anomaly detection can be shared between the nodes. However, communication is still required to communicate with the other nodes to reach to consensus. It is necessary for each node to be able to detect and recovery from failures independently while achieving the high recovery rate without incurring additional overheads. Although centralised approach may be easier to implement and can provide a higher accuracy, the nodes need to send information required for detection over the network. This can significantly increase the energy consumption and recovery time. Both centralised and hybrid approach can suffer from a single point of failure.

Table 2.1: Comparison between different fault tolerant architectures

| **Architecture** | **Centralised** | **Distributed** | **Hybrid** |
|---|---|---|---|
| *Agent Installation* | Sink | Node | Node (Detector) Sink (Responder) |
| *Response Time* | Increases as size of network increases | Real Time | Immediate detection only |
| *Communication Overhead* | High (Data) | Minimal (Alert) | Moderate (Alert) |
| *Processing Overhead in node* | Minimal | High (Detection) | Medium (Load sharing) |

## 2.6.2 Error Detection

Before any failure can be rectified, the error must first to be detected and the fault needs to be identified before recovery can be performed. Many error detections algorithms have been proposed in the literature to detect any anomalous behaviour in WSNs. Some of the proposed solutions integrate the detection with diagnostic or recovery as one system. Others treat them as an isolated problem, implement and test the error detection separately in different layer of the communication stacks. They assume that diagnostic and recovery will be provided by other system. Some of the key detection approaches to construct the normal model and detect the anomalies in the WSNs are discussed in this section.

### Type of Detections

Error detection can be categorised into two type namely *signature-based* and *anomaly-based* error detection (Chandola et al., 2009).

*Signature-based* detection is based on a pre-configured set of known error patterns that need to be updated when a new error pattern is found. It can only detect error based on the signature installed in the system. It cannot detect any new emerging fault. A new signature has to be created and deployed each time a new error emerges. Hence, signature-based error detection is less popular in WSNs domain due to the limited resources available in the nodes for communication and storage.

*Anomaly-based* detection, also known as Anomaly Detection Systems (ADS) is based on a normal model to detect any abnormal behaviour by monitoring any observation that significantly differs from the normal model. ADS initially constructs a model of the normal characteristics from an observed system. Using the normal model constructed, it can detect anomalies by observing any change in the current system behaviour. A system can be flagged as abnormal if the current model deviates from the previous normal model based on an acceptable threshold value.

ADS is suitable for the resource-constrained properties of the WSNs as it does not need to maintain and share the signatures for different errors. Each node can detect any observed activities that deviate from the normal behaviour during operation. However, the accuracy of the ADS depends on the precision on the capturing of the normal model, the anomaly threshold and the techniques used to distinguish between the norms from the abnormal. One major challenge in ADS is that it can be difficult to capture and establish the norms and define the acceptable threshold value especially with the dynamic normal pattern that changes with operating environment. This detection technique can be subject to a high false negative rate if the detection system is unable to observe and construct the normal pattern or model of the monitored system during initialisation or operational period for update. An ideal ADS in WSNs is a system that can achieve a high detection with no false detections with minimal resource consumption in a distributed streaming data.

### Establishing the norms

Models representing the normal and abnormal characteristics of WSNs need to be established and built using machine learning technique before anomaly detection can be carried. Chandola et al. (2009) classify the techniques to establish the model into supervised, semi-supervised and unsupervised learning approaches.

Supervised learning assumes the availability of labelled data sets to train the classifier (Görnitz et al., 2013). A supervised algorithm analyses the training data and constructs the normal and abnormal models using the labelled input data. A data point can be determined as normal and abnormal solely depending on which models the data point fits in. Many supervised learning techniques have been used in WSNs. Wang and Zhang (2007) have used statistical approaches to construct the node profile using the normal and abnormal traffic pattern. Hodge and Austin (2004) highlight that this approach requires both normal and abnormal data to be available. These pre-labelled data are usually not available and time consuming to obtain. The normal data in WSNs usually changes during operation. The available data might not be extensive enough to cover new and occasional events. It may require retraining if the normal and abnormal models change.

Semi-supervised learning combines a large number of unlabelled data with a few labelled data to improve classifiers (Görnitz et al., 2013). It declares any new data points that do not fit into the normal model as anomalies. When labelled data are limited, semi-supervised learning can be used to improve supervised learning using the readily available unlabelled data (Zhu and Goldberg, 2009). However, it is not easy to get a complete trained data set in the real world. Defining the boundary of normal can be difficult.

In unsupervised learning, all the data available to construct the classifier are unlabelled. It is used to construct the classifier online. It can detect the anomalies by applying various statistical methods and distribution functions to build the normal model based on the observed behaviour of the systems. It is common to assume that (i) the *occurrence* and the *population of normal data is higher than the anomalous data* as anomalies do not occur very frequently and (ii) *anomalies are distinguishable from the norms*. It requires a certain measurable criteria or threshold to identify the anomalies.

In summary, an unsupervised learning approach is applied in this thesis to monitor the time varying characteristics of the radio pattern experienced by the motes. The environmental normal radio pattern may change dynamically and is usually unknown prior to deployment. With the use of the unsupervised approach, the classification model can be built online according to the current temporal condition or when there is a change in the operating environment.

#### Error Detection Algorithms

Most of the ADS proposed in WSN in the literature detect anomalies based on periodic-based application using sensor measurement, RSSI, and network traffic

pattern as the input data. Chandola et al. (2009) have suggested that different detection algorithms proposed in the literature can be generally classified into five approaches namely Statistical-Based (SB), Classification-Based (CB), Nearest neighbour-Based (NB) and Cluster-Based (CLB). Table 2.2 summarises some of the existing ADS proposed in WSNs.

Table 2.2: Comparison of various anomaly detection algorithms

| DS | Anomalies | | Arc | Algorithm | Model | | Signature Update |
|---|---|---|---|---|---|---|---|
| | Type | Attribute | | | Learning | Training | |
| SB1 | Data | Single | DS | Statistical (Mean/SD) | Type-3 | Online | Static |
| NB | Data | Single | CS | KNN | Type-3 | Online | Dynamic |
| CB1 | Data | Single | CS | ANN | Type-1 | Offline | Static |
| CB2 | Data | Single | DS | SVM | Type-3 | Online | Dynamic |
| CB3 | Signal | Single | DS | BBN | Type-3 | Offline | Static |
| CB4 | Network | N/A | DS | Rule-Based | N/A | Online | Static |
| CLB | Network | N/A | DS | Clustering | Type-3 | Online | Static |

**Keywords**

***Detection Systems***

SB1 = (Hida et al., 2004)
NB = (Rajasegarar et al., 2008)
CLB = (Loo et al., 2006)
CB1 = (Kulakov and Davcev, 2005)
CB2 = (Rajasegarar et al., 2007)
CB3 = (Lee et al., 2007a)
CB4 = (da Silva et al., 2005)

***Learning Model***

Type-1 = Supervised
Type-2 = Semi-supervised
Type-3 = Unsupervised

***Arc = Architecture***

CS = Centralised System
DS = Distributed System

***Others***

ANN = Artificial Neural Network
KNN = K-Nearest Neighbour
SVM = Support Vector Machine
BBN = Bayesian Belief Network
SD = Standard Deviation
N/A = Not applicable

The majority of the ADS listed in Table 2.2 have the following key properties:

- *Abrupt vs. Subtle:* Most of the ADS have focused on malicious attacks and have not taken into account anomalies that occur within the operating environment, such as human-induced interference and wireless communication. The ADS proposed by Loo et al. (2006) and Hida et al. (2004) can detect abrupt attack but have not been tested on subtle and irregular changes that are more difficult to detect.

- *Distributed vs. Centralised:* In order to conserve energy, distributed approach has been taken to extend the lifespan of the network. It usually does not have a centralised controller and must work with localised data. All the nodes in the network can detect local anomalies but have not taken into consideration the temporal characteristics of the network (Loo et al., 2006). A weakness shared by the distributed approaches is that the ADS can only detect the anomalies based on the localised data. The local nodes may fail to detect the failure if when a large area of the local network is affected by the anomalies leading to similar behaviour being observed by the local nodes (da Silva et al., 2005). Although some centralised approaches are proposed to provide a global view, the data are usually compressed to reduce the communication overhead (Rajasegarar et al., 2006). As a result, data required to identify the anomalous nodes might be removed during data aggregation.

- *Static vs. Dynamic Threshold:* Proposed ADS requires a pre-defined threshold to detect the anomalies which can be very difficult to define prior to deployment (da Silva et al., 2005). It can be very difficult to determine an appropriate threshold value in the time variant WSNs and may need to be updated when the normal model changes. The boundary between normal and outlying behaviour is often not very clear (Rajasegarar et al., 2007). Therefore, defining a representative normal region is challenging. However, due to the limited constraint in the node, the building of the normal spatial and temporal model can be performed on-demand. This is usually triggered when a change in the operating environment is detected. Different anomaly metrics can be used to detect anomalies.

- *Online vs. Offline:* It may be necessary to generate the normal model online as the availability of labelled data for training/validation can be difficult to obtain prior to deployment in WSNs. These training data are usually available during or after deployment or with preliminary site survey. Capturing of the normal condition has to be performed online as the operating environment keeps on evolving. The data collected may contain noises that need to be manipulated in order to obtain the normal model of the data. Hence, techniques that required offline training may not be appropriate in WSNs. Due to the limited resource in the nodes, the online detection algorithm must not incur a high computational complexity. Techniques such as ANN (Kulakov and Davcev, 2005) and SVM (Rajasegarar et al., 2007) usually have a high complexity during the model generation making them unsuitable to be performed in the nodes.

In summary, the distributed detection should be performed near to the source of the anomalies to reduce communication overhead and recovery time. The algorithm used to detect the anomalous events online must be energy efficient due to the limited resource in the nodes. The detection model must be able to detect different types of anomalies and can adapt to the dynamic environment. Different anomaly metrics can be applied using different data attributes to improve the detection rate and identification of the anomalies.

### 2.6.3 Error Consensus

In WSNs, sensor nodes may be compromised or not functioning properly. Conflicting information generated due to these faulty nodes can be misleading and may cause the network to response incorrectly. This problem is commonly known as the Byzantine General problem and is a common problem in both distributed and hybrid architectures (Lamport et al., 1982). For example, the sink or CH depends on the alerts or data received from the sensor nodes to determine whether a node is malicious. If the information received is not inaccurate, the sink or CH will not be able to make decision or may take the wrong action that may aggravate the anomalous condition (Atakli et al., 2008). Currently, there are two methods to declare whether a node is compromised (Ioannis et al., 2007; Atakli et al., 2008). The first method proposed by Ioannis et al. (2007) is based on cooperative decision making where all the nodes will take the responsibility to perform the anomaly detection and send the alerts. A consensus on whether the node is compromised can only be made after all the alerts are received from the neighbouring nodes within the transmission range (Ioannis et al., 2007; Roman et al., 2006). A node is confirmed anomalous based on the majority votes from the neighbouring nodes. An alternative method is based on a trusted value assigned to an individual node where only nodes with a trusted value above a certain threshold is certified to send the alert to the base station to affirm the anomalies (Atakli et al., 2008). These trusted nodes are assigned by the base station based on a certain reliability condition of the data received from the trusted nodes. All consensuses require some exchange of messages between the nodes. Although these messages can piggyback on existing communication to reduce overhead, the messages can be easily be compromised or loss. Hence, a distributed self detection without relying on error consensus is necessary in WSNs.

### 2.6.4 Error Diagnosis

When an error is detected by the ADS, it is necessary to identify the location and the cause of the fault in order to isolate or recover the fault. This technique to assess the failures caused by the detected error is known as error diagnosis. An error diagnosis system needs to assess the effectiveness of the detection-recovery agent to rectify from the faults using a feedback mechanism. This feedback mechanism consists of information about the health status of the node or network after repair. With the availability of such system, the success rate of the recovery can be increased.

As discussed in Section 2.6.1, error diagnosis can be executed in the centralised and distributed manner. A centralised error diagnosis is performed in the sink or cluster head while distributed diagnosis is performed in the localised node. With the ability of the node to diagnose the fault, each node can decide and confirm independently on the state of the network without performing any consensus. You et al. (2011) classify the error diagnosis into proactive and passive diagnosis. In proactive approach, the diagnostic agent is embedded in the node to monitor the node's ability to perform its tasks such as routing and reporting the result to the sink periodically. Debugging software such as Sympathy constantly monitors the network to collect traffic statistics that is generated by each communicating node (Ramanathan et al., 2005). It uses a decision tree to locate and determine the cause of the failure based on the connectivity metrics obtained from the routing table, packet flow metrics and node status metrics. A fault is detected when the traffic from a node is not sensed for a certain period of time. This approach requires additional energy consumption for continuous monitoring and introduces a large communication overhead in the network due to constant reporting from individual node. The time window required to diagnose the network has to be set according to the metrics to ensure that the statistics collected can be used to locate and determine the cause of the failure. To minimise these overhead, Staddon et al. (2002) apply a centralised fault tracing approach to determine whether the cause of the network error is due to a single router failure or a group of nodes within an area. The sink broadcasts a route update to determine the status of the inactive nodes. Each node receiving the route update message will piggyback its neighbours IDs along with its reading to the sink. The sink can then process this information to learn about the topology of the normal network during training or learning mode. This normal network topology is then used to detect and trace the fault node during each routing update. Although the centralised approaches taken by Ramanathan et al. (2005) and Staddon et al. (2002) can provide a global view of the network and produce a higher accuracy, it is not

advocated for a large scale network as the information needs to parse through the network before reaching the diagnosis agent. It is very expensive to collect the information from every node and identify them in a centralised manner. A sufficient amount of network statistics has to be collected within a time window before a decision can be make on the fault nodes that may increase the mean time to detection. The centralised approach may also drain the energy level of the nodes closer to the sink and lead to a hot spot problem and network partitioning (Perillo et al., 2005). Hence, an autonomous distributed error diagnosis approach is required to address these problems.

### 2.6.5 Error Recovery

As the networks components play an important role in WSNs, it is critical to ensure that the network is always available for service and has the capability to tolerate failure. In order to prevent failure or bring the network back to its original state and maintain operational, it is necessary to rectify the erroneous condition immediately after an error is detected. In section 2.6.2, error detection can be detected in a distributed, centralised manner and hybrid manner. Error recovery can also exhibit the same architectures. If the error is identified and detected in the sink, the recovery request can be activated and instructed by the sink in a centralised manner. However, it takes a longer time to recover from the failed network as the information needs to be sent, processed and acted upon by the sink which can be distance away from the erroneous node. Individual loss packet may occur in noisy environments affecting the recovery process. These recovery requests may be loss along the route. As a result, the distributed recovery is more attractive in WSNs as errors can be recuperated by the nodes autonomously. Immediate response can be provided to rectify the failure immediately.

Different error recovery techniques have been proposed in different layers of the WSNs protocol stack such as congestion control in the transport layer, path recovery in the network layer and collision recovery in the MAC layer. The following subsections discuss each of the approaches.

#### Congestion Control

Wan et al. (2003) propose local congestion detection in the transport layer of the node. Each node broadcasts a message back to the sources node to reduce the congestion rate in the local network when congestion is detected. The node detects congestion by sensing the channel. Each node receiving the message will adjust it sending rate according to the local congestion policy to overcome the

congestion. As a node will continue broadcasting the messages as long as congestion persists, this may aggravate the congested network as the broadcast messages propagate toward to the source node. To reduce the communication overhead, Sankarasubramaniam and Akan (2003) propose the Event-to-Sink Reliable Transport (ESRT) to manage the congestion by adjusting the reporting rate at a sufficient level required for the sink to reliability to detect the event. A desired reliability has to be defined and the observed reliability must be within a tolerance range. If the reliability is below the predefined range, the sensing rate will increase. ESRT can minimise the communication overhead and control the traffic flow provided if the network is free from other interference sources. ESRT can suffer from high latency and packet loss in a multihop network as it relies on end-to-end feedback between the sink and the source. To address this long traversal of feedback messages, a hop-to-hop reliable transport protocol such as Pump-Slowly, Fetch-Quickly (PSFQ) can be applied (Wan et al., 2005). However, PSFQ does not perform any congestion control.

A hybrid approach is proposed by Rajendran et al. (2004) that use the local queue in the nodes to determine the network congestion. Each node maintains a neighbourhood congestion table to keep track on the cause of the packet dropped. Reliable Adaptive Congestion-controlled Transport protocol (ReACT) combines source-based congestion and error control with receiver-initiated localised recovery to notify the downstream nodes the occurrence of congestion. A local or global recovery is then activated depending on the congestion status of the neighbouring nodes. The local recovery attempts to recover localised losses such as transmission errors while the global recovery is trigger for errors that could not be rectified locally. ReACT has the ability to distinguish between local and network losses using the MAC layer information and can quickly correct errors and path loss with lower overheads when compared to an end-to-end protocol.

**Path Recovery**

There are two basic approaches used to redirect traffic during path failure namely table-driven or on-demand rerouting. Both of these rerouting approaches can be performed globally or locally. The recovery decision on the route taken can be made based on the resource or link availability. When the decision made is based on a particular network element such as node battery level, it is a resource-oriented decision. Whenever the recovery mechanism is focused on a particular path based on the link quality, it is considered to be path-oriented.

Most routing protocols use the cumulative link statistics between nodes to detect failures. If a sensor node cannot receive packets from a neighbouring node,

the node is classified as a failed node and removed from the neighbour table. The next best node can be selected from the table as the next-hop node. This type of route recovery is known as table-driven rerouting. In table-driven rerouting, multiple routes are established in the node during initial route discovery in order to provide the alternative route during failure. Marina and Das (2002) apply a multiple paths approach to forward the packet to tolerate a next hop failure. A backup rerouting can provide an alternative route with minimum delay. However, the process to manage and maintain the list of the backup nodes may require additional overhead. The backup nodes can also be subject to fault. As a result, an on-demand route discovery is more suitable to determine an alternative route during network error. Local recovery is attempted by the upstream node to protect against a link or node failure (Perkins and Royer, 1999). If the local node is unable to recover a failed path due to unavailable of redundant node or unsuccessful attempt, a failure message may be send to the source where a global recovery is triggered to determine a new route. Global recovery is usually slower than local recovery and is usually applied when a local network has failed.

**Collision recovery**

Collision is a common phenomenon in networking, in particular in a shared wireless channel. WSNs protocols such as flooding and data aggregation are especially prone to collisions. Most of the collision recovery approaches in WSNs are based on collision avoidance and prevention approaches. Duty cycling has been applied to reduce the occurrence of collision (van Hoesel and Havinga, 2004). Different radio channels are used by different clusters of nodes to reduce the interference (Heinzelman et al., 2000). However, such approaches usually rely on the topological structure of the sensor network and apply in a hierarchical structure. They also require redundant nodes or the ability of the node to adjust or synchronise its radio to perform the duty cycling. In channel or frequency switching (Song et al., 2008), a consensus is required to decide on the channel reservation between different clusters that may generate additional overheads.

Collision detection is important in WSNs before collision avoidance schemes such as RTS/CTS can be applied. Collision detection is performed by channel sensing (Ye et al., 2002). Whitehouse et al. (2005) exploit the capture effect to detect and deter from packet collision caused by flooding. The capture effect is defined as the ability for a receiver to sense and receive a signal from one transmitter despite the interference from another transmitter, even if the relative signal strengths of the two signals are nearly the same (Leentvaar and Flint, 1976). A node listens for any packet preamble during packet reception to detect for colli-

sion. If the signal strength of the second packet is stronger than the current packet the node is receiving, the node can stop receiving the first packet and resynchronise in order to recover the second packet. The capture effect can differentiate between a high collision rate and low link quality. It can decide whether to change its transmission schedule or to find a route. If successive retransmission is unsuccessful, an adaptive power transmission algorithm is applied to overcome the interfering signal (Lin et al., 2006). The capture effect can only detect and recover from collision created by a second packet with stronger transmission strength. The accuracy of the detection may decrease if the time difference between transmissions decreases as the second packet is received during or at the same time as the first packet. One approach to recover from collision is to apply a transmission control to change the transmission schedule or perform a back-off to change its next transmission phase (Woo and Culler, 2001). For example, the CSMA/CA in TinyOS uses retransmission and exponential back-off algorithms to recover from packet collision (Gutierrez et al., 2001).

### 2.6.6 Limitation of Existing Fault Tolerant Approaches

Many fault tolerance algorithms have been proposed that differ accordingly to the information used for analysis and the techniques applied to detect deviations from normal behaviour. Unfortunately, little work has been done with respect to anomalies with subtle and irregular change caused by environmental effects, with majority of the solutions focus on sudden change caused by faulty or malicious nodes. Most proposed algorithms have the following limitations:

- ***Adaptability:*** The network detection model or signature is usually generated during initialisation period where the network is assumed to be free from anomalies. This model remains static during testing and does not change as the networks evolve. The behaviour of WSNs usually changes as the system operates due to buffer overflow, route expiry, and a change in the operating environment. The detection model should adapt to the time-varying wireless channels and the variations of the network topology. For instance, the communication protocol must adapt its responses according to the current state of the networks.

- ***Limited recovery mechanism:*** Most proposed solutions have treated recovery as a separate issue and tested separately not as a whole system (Schaust and Szczerbicka, 2011). Some of the typical recovery strategies involve rebooting, node replacement, adaptive transmission range, or using mobile

node to cover the transmission range temporarily. These solutions may incur temporary network outage while the faulty node is being repaired. To separate recovery from detection and diagnostic would be impractical in the resource constraint WSNs where each process has to be executed separately.

- ***Cross layer design:*** Cross-layer interactions have been used to obtain the state of the node and to optimise the protocol. Existing network recovery approaches such as congestion control and error recovery are implemented in the transport layer and rely on the information provided by the network or MAC layer. For example, ESRT requires the link quality information to detect the congestion or collision from the MAC layer to perform the congestion control. Routing recovery also relies on this information to detect route failure before any recovery approach can be initiated. This information are passed between layers and processed in the layer to perform the recovery. This will require additional resources for processing, storage and communication which are limited. It is necessary that the fault tolerance to be performed in the same layer to minimise the cross layer interactions and avoid additional resources consumption.

In summary, it is crucial that the fault tolerance for the WSNs is distributed and lightweight to reduce energy consumption. Unsupervised learning approaches are usually used to generate the normal model as data are usually not available prior to deployment. Dynamic detection model are required to capture the spatial and temporal variants of WSNs. In the next section, we explores the application of immune-inspired approaches to provide fault tolerance.

## 2.7 The Application of Immune-Inspired Approaches toward Fault Tolerance

In Section 2.6.6, we have highlighted that the current statistical approaches are not sufficient to address the non-linearity behaviours of the WSNs and an adaptive approach is required. As a result, researchers have explored on the immune-inspired approaches for anomaly detections (Wallenta et al., 2010; Drozda et al., 2011). The immune system is a unique complex defence system that protects the body from different types of foreign micro-organisms present in the environment (Goldsby et al., 2003). It can identify and eliminate specific micro-organisms invading the body. These micro-organisms, also known as pathogens, can be harmful and trigger an immune response when expose to the immune cells. According

to Wallenta et al. (2010), the immune systems are applicable to the WSNs as they exhibit similar properties as WSNs (summarised in *italic*) listed in Section 2.2 such as:

**P1:** ***Resource Constraints***

- *The WSNs consist of sensor nodes that have limited resources.*
- The immune system is made up of specialised cells that have limited processing and storage and communication capacity. Each cell can bind and communicate with a restricted number of neighbour cells.

**P2:** ***High density***

- *The WSN's nodes are usually deployed in large number.*
- The immune cells exist in large number.

**P3:** ***Uncontrollable operation***

- *The nodes prone to failure as their radio are subject to interference, fading and contention.*
- Each cell is prone to failure as it operates in an environment that is continuously exposed to attack.

**P4:** ***Distributed autonomous***

- *Each node is usually configured to perform multiple tasks in a distributed and autonomous manner without a centralised controller.*
- Each cell performs different tasks in a distributed autonomous and self-organising manner without predetermined global control.

### 2.7.1 The Immune System

To defend and protect the body from harmful pathogen, the immune system uses a multi-layer approaches that is partitioned to four layers namely the outer barrier, biochemical barrier, innate immune system and adaptive immune system (Goldsby et al., 2003; Murphy et al., 2012). The outer barrier, such as the skin, provides the first layer of protection to prevent harmful pathogen from entering the body. The biochemical barrier consists of destructive enzymes that inhibit the growth of micro-organisms and eliminate them. The innate immune system reacts to pathogens that managed to break through these barriers. It can recognise common and frequently encountered pathogen that are not specific to any type

of pathogen. The adaptive immune system can detect and learn new and specific pathogen. The activation of each immune system depends upon the activity between the immune cells and the pathogens. To select an appropriate immune function for self-healing, it is necessary to understand the functions of the immune systems in more details.

**Innate immune system**

The innate immune system provides the initial protection from infections. It has the ability to recognise and response immediately to a foreign cell entering the body. It consists of different specialised white blood cells, such as macrophages and dendritic cells that can engulf and destroy the foreign cells. These cells also generate molecular signals to attract and interact with other immune cells and molecules. These molecules play a major role in fighting infections and removing dead cells to regenerate healthy tissues. They also interact and initiate the adaptive immune response using the specialised cells called the antigen presenting cells (APCs) (Murphy et al., 2012). The APCs process the antigens and present them to T-cells that may result in a secondary immune response.

**Adaptive immune system**

The adaptive immune system is a natural defence mechanism that is only present in vertebrate mammals. It has the ability to identify and remove foreign cells that are not recognised by the innate immune system. The lymphocytes are a group of white blood cells in the adaptive immune system that have the ability to change their molecular structure dynamically. These cells are initiated and regulated when the antigens on the surface of the APCs are presented by the innate immune system (Goldsby et al., 2003). The lymphocytes are produced in the bone marrow through the differentiation of stem cells. These stem cells can differentiate into specific and non-specific lymphocytes. Specific lymphocytes learn and recognise specific antigen using various receptors (protein molecules) on their surface and attach these receptors to the receptors of the antigen. There are two types of antigen specific lymphocytes produced in the bone marrow namely B Lymphocytes (B-Cells) and T Lymphocytes (T-Cells) (Janeway Jr, 1992). The B-cells mature in the bone marrow while T-Cells are released from the bone marrow and transported to the Thymus for maturation. Both cells express unique antigen-binding receptors on their surface known as the B-Cell Receptor (BCR) and T-Cell Receptor (TCR) that has the ability to bind to antigens and activate the cells (Novak et al., 2006). The activation of B-Cells produces antibodies that can

recognise antigens and responsible for effector function. The activated T-Cells regulate the response actions undertaken by other immune cells to destroy the infectious cells.

### 2.7.2 Immunological Theory

In order to prevent the immune system from attacking its own cells, it is critical for these immune cells to be able to distinguish between its own cell (self) from foreign cells (non-self) and attack only non-self cells. There have been many immunological theories that describe the discrimination process of the immune system.

The self/non-self discrimination produces cells that are capable of detecting non-self cells in the body through the process of positive and negative selections (Janeway Jr, 1992). The positive selection selects the lymphocytes with the receptors that can recognise and react with antigens while the negative selection eliminates all the lymphocytes that react with self antigens. For example, immature T-Cells recognising a self-antigen after positive selection process will undergo negative selection. Any T-Cells match a self-antigen will be removed leaving cells that can only be activated by a non-self antigen.

In contrast, the clonal selection principle (CSP) describes how the lymphocytes can evolve into effector cells when exposed to antigens (Burnet, 1959). When a B-Cell is exposed to a non-self antigen together with a co-stimulatory signal from the T-Cells, the B-cell is activated and reproduced. In order to diversify the cells colony and increase the probability of recognising more antigens, each newly generated B-Cell undergoes somatic mutation to become memory cells and plasma cells that can secrete antibodies with a higher binding strength. The plasma cells also undergo negative selection to ensure only cells reactive to non-self antigen are released into the blood stream. However, negative selection is not a perfect process as it cannot recognise every antigen with sufficient certainty (Burgess, 1998). As a result, self-reactive T-Cells and B-Cells can be released to the blood stream to attack the body.

In Jerne (1974) immune network theory, the immune cells are capable of interacting and stimulating each other without the presence of on antigen and are always in a state of dynamic equilibrium. Jerne (1974) proposes that the antibodies is capable of recognising other antibodies and can bind to the surfaces of other antibodies forming a network of antibody-antibody interactions. These interactions can generate positive and negative response. A positive response causes B-cell activation, proliferation and generation of antibodies. A negative response leads to cell tolerance and suppression to prevent autoimmune response.

Due to the non-reactive feature of immune cell to good bacteria and autoimmune disease cannot be explained simply by self and non-self discrimination, Matzinger (1994) suggests that the immune system can detect and respond to anything that is dangerous. This immunological principle is known as Danger Theory (Matzinger, 1994). In Danger Theory, specialised APCs called Dendritic cells (DCs) are released to the body to initiate the immune response. They collect antigens from foreign and host cells in tissues, and present these antigens to naive T-cells. The DCs operate in three states namely: immature, semi-mature and mature. During normal operation, immature DCs will differentiate into semi-mature state when they receive signals generated by T-Cells due to natural cell death. Semi-mature DCs will suppress the matching T-Cells. However, if the T-cells are injured, stressed or induced death, danger signals are released requesting for an immune response. Upon detecting the danger signals, immature DCs differentiate into mature DCs. These mature DCs release a number of protein molecules to stimulate and activate the T-cells for an immune response.

Immunological principle proposed by Grossman and Paul (2001) postulate that the lymphocytes have the ability to tune and update their responses according to its current environment. The immune system can recognise and classify different patterns of signal received and trigger selective responses based on the current on-going activities with other cells. The activation of the T-Cell does not depend only on the type and density of the foreign bodies, but also on the signals received from other cells. For a T-cell to be activated, the TCRs must be stimulated by an APC until the excitation level exceeds an activation threshold. Recent work by Owens et al. (2010) show that the activation of the T cell to discriminate between the self and non-self antigens depends on the stimulation received from all the antigens and the internal signalling of the T-Cell. The internal component of the TCR undergoes a process known as kinetic proofreading that has the ability to adapt the activation of the receptors (Owens et al., 2010).

In Summary, the immune system is an immensely rich and complex natural defence system that has the capabilities to learn new disease, recognise previously encountered micro-organisms and adapt the system to the new environment. The properties of the immune system to trigger an immune response in an autonomous and distributed manner have provided the inspirations for the development of immune-inspired algorithms to solve real world complex problem especially in anomaly detection. Although there are many conflicting theories on the activation of the immune response, various AIS algorithms have been proposed and tested based on these theories. Some of them have been successfully implemented for error detection in swarm systems such as WSNs and robotics

and have produced promising results (Lau et al., 2011). The next section introduces the AIS algorithms and investigates some of the proposed AIS applied in fault tolerant system and WSNs domain.

### 2.7.3 Artificial Immune System

De Castro and Timmis (2002) define AIS as *"the adaptive computational systems, inspired by the theoretical principles and processes of immune function, used for problem solving"*. AIS exploits the immune system's adaptability, learning and remembering capability making them appropriate to appropriate to solve the complex problem in WSNs and apply the computational and mathematical approaches of immunity to solve complex problems. It attempts to algorithmically mimic the behaviour of natural immune system. Over the years, different AIS algorithms have been proposed and applied in WSNs due to its ability to self-organise and adapt to the environment. The properties of the WSNs made the WSNs a suitable metaphor for the tissue of the human body (Wallenta et al., 2010). We will describe the algorithms proposed in the following subsections.

#### Negative selection algorithm

Negative selection algorithm (NSA) is an algorithm inspired by the maturation of the T-Cells in the thymus where each cell undergoes a selection process to form a set of mature T-Cells that can only bind to non-self antigen. Forrest et al. (1994) apply NSA to detect any infected file in a computer system. A set of signatures is produced by removing the data that matches a set of self strings during training. Any incoming data instance that matches any detector will be declared as anomalous during testing. Hofmeyr and Forrest (2000) later extend the NSA to detect network intrusion in a wired network. They classify the network connections into binary string of self and non-self and introduce a permutation mask to reduce the number of holes and the false negative rate. These holes occur because the matching techniques and strings used cannot cover all the non-self detectors (D'haeseleer, 1996). Le Boudec and Sarafijanovic (2004) implement the NSA in mobile wireless network to detect abnormal nodes using the routing traces collected from neighbouring nodes that is later implemented in WSNs to detect anomalies (Drozda et al., 2007). The NSA algorithm suffers from severe scaling and coverage problem. The time taken to generate an appropriate number of detectors can be very long as the search space increases. Kim and Bentley (2001a) estimate that it may take thousands of years to produce detectors set that can

only product 80% detection rate. The NSA is not suitable for generating competent detectors but is good for removing invalid detectors. Stibor et al. (2006) show that the performance of NSA is not better than the kernel density and SVM for detecting anomalies and can suffer from the curse of dimensionality where the number of samples required for classification increases exponentially with the data dimension. Elberfeld and Textor (2011) resolved this issue by proposing the string-based negative selection algorithms to reduce the worst-case execution time complexity from exponential to polynomial with the use of data compression techniques.

#### Clonal selection algorithm

Clonal selection algorithm (CSA) is based on the proliferation and mutation process of the B-Cells as outline in section 2.7.2. The first CSA, CLONALG is used to perform pattern recognition and optimisation function (de Castro and Von Zuben, 2002). de Castro and Von Zuben (2002) demonstrate that CSA can select, reproduce and mutate a set of artificial B-cells from a set of input patterns. Kim and Bentley (2001b) apply the CSA to generate a set of detectors to detect network traffic anomalies. The input data are classified into self and non-self and compared against a randomly generated detector. If a detector can match a number of non-self antigens, it will undergo clonal selection to reproduce a population of detectors using crossover and mutations process. To ensure the validity of the detectors, the detectors undergo negative selection to remove self detectors. The algorithm was tested using breast cancer data and the results have shown a high false positive rate as the size of detectors increases. Kim and Bentley (2002) further extend the CSA to dynamically administer the population of the three detectors into immature, mature and memory detector. An additional external co-stimulation signal, based an acknowledgement from system operators, is introduced to verify the anomalies. Their results have shown a significant reduction in false positive when the normal remains unchanged. However, the false positive rate increases when the algorithm is applied in a dynamic environment where the self changes with time.

#### Immune network model

de Castro and von Zuben (2000) have proposed the Artificial immune Network (AiNet) which is a modification of the CLONALG algorithms with an additional feature to suppress the antibodies interaction. Fang and Lin (2005) have proposed an unsupervised anomaly detection algorithm based on AiNet on detect network

data anomalies. It is based on a network of antibody components to adjust and match a number of inputs. The algorithm uses AiNet to reduce the population of the input data by applying a hierarchy clustering algorithm to generate clusters of normal data. A cluster is normal if the number of data points in a cluster is more than 10% of the training data. A new data point is considered as anomalous if the distance from data point to a cluster is more than a predefined threshold. They highlight that detection rate of 85% can be achieved if an appropriate affinity threshold is used. However, the algorithm is not able to detect anomalies if the anomalies are not qualitatively different from the normal instances. There are many tunable parameters used in the algorithm and the sensitivity of these parameters on the detection rate has to be tuned to reduce false positive rate.

**Danger model**

Aickelin et al. (2003) proposed the first comprehensive application of danger theory in computer systems where a computational model of danger theory is presented and a novel ADS algorithm is proposed. The ADS system collects signals from the network and correlates the signals with intrusion alerts. Once a number of successful attacks are detected, a danger signal will be generated. The system requires an accurate classification of good and bad alerts which can be difficult to determine. Greensmith et al. (2005) extend the danger theory and propose the Dendritic Cell Algorithm (DCA). The DCA is based on the ability of the DCs to activate the T-cells for an immune response by evaluating the concentrations of both internal and external signals received (Mosmann and Livingstone, 2004). The external signals known as pathogen associated molecular patterns (PAMPs) are only produced by foreign micro-organisms. The immune responses mediated by the T-cells depend on the level of internal signals emitted by the tissue cells of the body. These tissue cells secrete chemical messengers known as cytokines to communicate with each other. By differentiating the cytokines generated from the tissues, pathogenic cell can be detected Aickelin et al. (2003).

**Tunable Activation Threshold Model**

Recently, a new AIS algorithm inspired by the Grossman and Paul (2001) T-cell signalling processes in TAT has been developed by Owens et al. (2012). Grossman and Paul (2001) postulate that the activation of a T-Cell depends not only on the excitation and suppression signals experience by the T-cell, but the excitations (exposure to the APC) must be stronger than the suppression to exceed the activation threshold. A T-cell that receives continuous interactions with the

same antigens will have a higher activation threshold. As a result, more excitations are required to exceed the threshold. The T-cells that interact regularly with self-antigens will also have a higher activation threshold (Grossman and Paul, 2001). This immunological theory has provided inspirations for development of anomaly detection algorithms. One such algorithms is the Receptor Density Algorithm (RDA) developed through the extraction of features of the generalised T-cell receptor, and mapped onto the kernel density estimation (Owens et al., 2012). The RDA is specifically designed for anomaly detection problems and has been applied in chemical and fault detection. Hilder et al. (2012) have shown that the RDA can successfully detect and distinguish different chemical substances in a dynamic environment and have been tested in real sniffing robot. RDA has also been tested on a swarm of robots to detect faulty components (Lau et al., 2011). Their results have shown RDA can achieve a higher detection rate with low positive rate compared to conventional statistical approaches.

### 2.7.4 Application of AIS in WSNs

Over the years, many immune-inspired ADS have been applied to different application areas of WSNs such as data classification, anomaly detection and coverage problem. This is partly motivated by the analogy between the characteristics of WSNs and the immune system. Davoudani et al. (2007) have provided a mapping between the Cohen's Cognitive Immune System (CIS) and the WSNs and have highlighted that the functionality challenges faced by WSNs are similar to those faced by immune systems where each node (cell in immune system) needs to maintain and regulate its operation as long as possible to meet the application requirements. The ability of the immune system to self-heal and adapt to changes in the environment have attracted the application of AIS in the area of anomaly detection. The earliest work is carried out by Ishida (1997) where they have used immune network theory as metaphor to detect sensor fault. Their solution is based on the B-Cells interaction to create a dynamic sensor network where distributed individual nodes can interact with one another to detect any inconsistent change. Wallenta et al. (2010) evaluate the DCA using TOSSIM and J-Sim simulator with 10 and 7 nodes respectively and have shown that DCA can only detect 65% fake interest packets. The detection relies on the PAMP signal usually generated by the sink to confirm the presence of the anomalies. This centralised approach is not practical in WSNs as the size of WSNs can scale up to thousand of nodes and may incur high communication cost. In real application scenario, the sink can be several hops away. The delivery of PAMP signal can be very expensive, or dropped due to congestion and collision. As a result, DCA is subject

to single point of failure if the sink is unable to validate the attack. The DCA solution also assumes the WSNs logical topology does not change.

Similar to the technique proposed by Le Boudec and Sarafijanovic (2004), Drozda et al. (2007) apply the NSA to detect abnormal packet drop in WSNs on individual node. During training, randomly generated antigens (based on various network traffics) that do not match a set of self strings will become the set of mature detectors and use in the anomaly detection by the node. The simulated results collected from GlomoSim simulator have only shown 70% to 85% detection rate as inadequate number of detectors were generated during training. To improve the detection rate, Drozda et al. (2011) apply the interaction between innate and adaptive immunity to classify the errors that can lead to degradation in packet delivery rate. Co-stimulation signals between the innate and adaptive immunity are used to reach to a consensus on the presence of the fault. Schaust and Szczerbicka (2011) propose an automated response system based on the CIS in order to ensure that the networks are always available for service. Detection without effective diagnosis is not sufficient to determine the underlying cause of the problem, and confirm the presence of the fault for an appropriate remedial action to be taken. These three actions have to be integrated and operate as one component. In addition, for all these studies, the proposed solutions mainly focused on failure caused by malicious attacks. The proposed AIS algorithm by Davoudani et al. (2007) has limited adaptivity and the hybrid approach may require additional communication for interaction which can be expensive in WSNs. They are not suitable for the complex and dynamic environment of the WSNs as they does not provide online unsupervised learning and adaptivity required to detect changes in the radio environment. These attacks usually have unique features that are easier to detect and does not change dynamically.

### 2.7.5 AIS for Interference Recognition in WSNs

One of challenges in detecting anomalies due to interference is that the duration and occurrence for these types of anomalies are unpredictable and changes with time (Liu et al., 2010). According to (Boers et al., 2010), it can be expensive and difficult to detect and classify the presence of interference using existing AIS or other statistical approaches such as computing the *means* and *median* discussed in Section 2.6.2. Detecting anomalies using such traditional statistical approaches may only detect anomalies that have a high variability. The results from Boers et al. (2010) has shown that the statistical techniques by taking the mean and skewness of the RSSI values cannot be used to distinguish between different interferences. The duration and occurrence of these interferences are usually dependent on the

type of radio devices or applications, and their usage pattern. Due to the limitation of current statistical approaches and the non-adaptive of existing AIS algorithm to changes in the environment, this thesis will investigate the application of an AIS algorithm based on the signalling of the T-Cell receptors to support the nodes adaptivity to the dynamic radio environment. As discussed in Section 2.7, the T-Cells have the ability to adapt the activation of their receptors according to the recurrent signals received on each receptor. The stimulation of the cell's receptor must exceed the activation threshold in order to be activated. The abilities for the cell to *act and detect independently* from other cells without incurring additional communications and *adapt the activation* of its receptors according to its current environment have motivated the application of the RDA in WSNs node to detect the presence of the irregular radio interference in the WSN's environments. Based on the numbers of activated receptors, the radio interference can be determined as the receptors will be activated according to the strength and frequency of the interference captured. With these information, an appropriate response can be taken.

## 2.8 Fault Tolerance in WSNs Routing Protocols

The routing protocol plays an important role in order to achieve dependability as data packets are often routed across the WSNs according to the routing algorithm to reach the final destination. The routing protocol must be resilient and can react to topology changes stemming from unreliable links or faulty nodes to ensure the stability of the network infrastructure under varying network dynamics. The ability to detect and recover from changes or failures is necessary. The routing protocol can be administered proactively, reactively or both. Al-Karaki and Kamal (2004) highlight that it is preferable to have proactive routing protocols rather than reactive protocols when the node is static. However, WSNs have dynamic topologies that may change according to its operating environment even if the nodes are static. Reactive routing is known to be fault tolerance to topological change as it allows the nodes to react quickly to failures (Al-Karaki and Kamal, 2004). Hence, in this section, we focus only on the reactive routing protocols to identify the detection and recovery approaches to overcome failures.

### 2.8.1 Reactive Routing in WSNs

The two reactive protocols widely researched in WSNs are the Adhoc On-demand Distance Vector (AODV) proposed by Perkins and Royer (1999) and Dynamic

Source Routing (DSR) proposed by Johnson and Maltz (1996). Both AODV and DSR initiate a RD to establish the route to the sink using two main routing management packets namely: Route Request (RREQ) and Route Reply (RREP). When a source node has data to send to the sink, the source node broadcasts a RREQ packet. Each RREQ packet contains an unique sequence number, the source address, the destination address, the route path and the time to live (TTL) counter. When an intermediate node receives a RREQ packet, it checks the sequence number and inspects the TTL counter in the packet to prevent re-broadcasting and mitigate loop formation that may lead to broadcast storm (Tseng et al., 2002). Broadcast storm can severely affect network performance and disrupt the network communication. When the RREQ packet is received by the sink or a routing node that has a route to the sink, a RREP packet is sent to the requesting node via the reverse path the taken by RREQ packet. Once the source has received the RREP packet, the source node can begin to send the data packets to the sink.

Abolhasan et al. (2004) highlight the two differences between AODV and DSR. Firstly, AODV uses a routing table to forward the packet to the next hop neighbour. Each node relies on the destination address and the next hop address in the packet's header to forward the packet. The next hop is determined using the routing table computed during RD (Perkins and Royer, 1999). As for DSR, the data packet carries the routing information on its packet header that contains a list of forwarding nodes (Johnson and Maltz, 1996). Each routing node will inspect the forwarding list and forward the packet accordingly. This list can become very long if the distance between the sink and source is far away. As a result, the routing overhead in DSR is higher than AODV due to a larger header size. Secondly, DSR does not support local route repair in an event when one of the routers fails (Youn et al., 2006). The route is repaired based on the routing information that is cached in the source node. The route information is updated when an upstream node detects a failure and notifies the source by sending a route error (RERR) packet. In contrast, AODV repairs its route locally and is more adaptable to highly dynamic environment compared to DSR. As a result, DSR is less tolerant to failure in a large network and is not analysed in our work.

### 2.8.2 Mixed Routing

Alternative hybrid routing that combines the proactive and reactive routing protocols has been proposed in various literature to reduce packet delay and improve packet reliability and energy efficiency. The Zone Routing Protocol (ZRP) proposed by Haas and Pearlman (2001) is the first hybrid routing protocol to be implemented in WSNs to reduce the routing traffic. Proactive routing is applied

within a cluster of nodes to manage the local route and reactive routing is performed between clusters. Although the ZRP can utilise the wireless spectrum effectively, it is unable to adapt to the dynamic topological changes and therefore it requires the cluster to be restructured. A policy-based approach has been proposed to adapt to the dynamic topology of WSNs (Figueredo et al., 2005; Yazir et al., 2010). Figueredo et al. (2005) proposed a policy-based adaptive routing in WSNs where a set of nodes can switch between reactive and proactive routing protocols depending on the forwarding policy. These policy-based approaches use a centralised routing decision determined by the destination node using the network delivery rate observed and a threshold level. However, the centralised approach is subject to single point of failure and can incur additional communication overhead to forward the policy. Yazir et al. (2010) propose a multiple criteria analysis method for a decentralised decision making where a distributed routing decision is made locally based on aggregated voting between a group of local nodes. As a result, additional delay and overhead are required for the distribution mechanism to reach for a consensus on the routing decision and for the network to stabilise. Due to the difference in reactive and proactive protocols, the routing switching module may cause service discontinuity when individual node switches between the reactive and proactive routing (Figueredo et al., 2005; Yazir et al., 2010). The nodes need to reset and reconfigure their routing service and allow the same routing protocol to run in all the nodes. In order to reduce the switching downtime. the spectrum of routing protocols applied must exhibit route management and recovery strategies, for instance, a combination of different variants of reactive routing protocol. In these thesis, we will investigate the applicability of integrating different variants of AODV to provide different recovery strategies.

### 2.8.3 Research Attempts to Enhance and Augment AODV

AODV presents several problems that are related to high packet error rate and high routing overheads created by lossy radio link (Pirzada and Portmann, 2007; Alshanyour and Baroudi, 2008). These problems can cause packet losses, packet collisions, and high end-to-end delay. Fault-tolerant AODV has been proposed to address these problems through redundancy. Researchers have extended and modified the AODV routing protocol to improve its performances in term of reliability and energy efficiency and the ability to provide an alternative better route (Marina and Das, 2002; Carbajo et al., 2008). This section surveys some of proposed enhancements to improve the AODV routing protocols in the literature.

- **AODV:** AODV is one of the commonly used reactive routing protocols in WSNs originally proposed by Perkins and Royer (1999) for Mobile Adhoc Network (MANET). As discussed in 2.8.1, it uses RREP, RREP and RERR to construct and repair the route from the source to the destination by initiating a RD. Although hello packet is proposed in AODV to maintain the link, MAC layer Link Layer Notification (LLN) is usually applied to reduce the communication overhead. When the AODV fails to receive a LLN, the route is repaired locally. If the route cannot be repaired locally, the packet is dropped and a RERR packet is transmitted back to the source. When RERR packet is received, the source will initiate a global RD to determine the route to the destination. Packets transmitted during RD usually have a higher latency as the networks are usually more congested than usual while the route is being established. In WSNs, spurious link failures or packet drops are common. This can be mistaken as failure and RD may be executed. Frequent RD can consume significant amount of resources and can reduce the lifetime of the network. It can also aggravate the congested network.

- **TinyAODV:** TinyAODV (Carbajo et al., 2008) is a minimalist adaptation of AODV used in WSNs node, such as telos motes, running TinyOS. It is developed based on pessimistic approach to recover from route failure without local repair and expiration period in the routing table. The routes are kept in the nodes when they are constructed during RD. When a packet cannot be sent over an active path, RERR is generated. The routing table is updated and the undelivered packet is dropped. It assumes that the route is unrecoverable and a global RD is required to determine the new route. It also assumes a static WSNs topology with minimum link failure.

- **AOMDV:** In Adhoc On-Demand Multipath Vector Routing protocol (Marina and Das, 2002), a multiple link-disjoint path is obtained and calculated from source to destination during global RD. Each node in the network will keep a list of alternative hops that is established during global RD. This list of alternative next-hop is ranked according to the hop-count. During a link failure, an immediate node will select the next available next-hop in its routing table to forward the packet. However, the list of backup route is not updated to reflect any changes in the network. As a result, this may lead to outdated or stale route. If all paths to the destination fail or no alternative path is available, RD will be initiated. This protocol can handle any route failure if the failure size is small. However, additional memory overheads are required to store multiple routes information. The use of *hello* packet is

needed to maintain the connectivity to all the neighbours, generating additional communication overheads.

- **NST-AODV:** Not-So-Tiny-AODV (NST) (Gomez et al., 2006) is an enhanced AODV that maintains all of the features available in AODV. It uses an optimistic recovery approach where the route failure is temporary and can recover by retransmission. In order to minimise the routing packets generated during by local repair, it performs an additional network layer retransmission to overcome sporadic failure caused by bad radio connection. Gomez et al. (2006) have reduced the network latency and the number of routing packet transmitted.

**Route recovery approach**

From the survey conducted, it can be concluded that these protocols are designed to handle specific network conditions. Table 2.3 highlights the operating environment that is suitable for each routing protocol. Different WSNs protocols may behave differently when applied to different network conditions. A specific routing protocol may perform better in one network scenario but can aggravate the network when applied in different scenarios. For example, the proposed AOMDV may fail to forward the packet using the backup route if all the neighbouring nodes are spatially interfered by a strong radio signal. The use of RD and hello packet may congest the network making normal communication unfeasible. When a link failure is detected, the node detecting the failure will attempt to restore the connectivity locally. This process is transparent to the source or sink. It is employed by most of the routing protocols discussed above except TinyAODV, where a pessimistic approach is taken where no local route is assumed to be available and a new route needs to be discovered globally. This pessimistic approach may not be able to tolerate anomalies in large network where frequent intermittent link failures can cause uncontrollable amount of routing packets being generated from the source. In an attempt to reduce the amount of control packet, AODV and AOMDV and NST use layer 2 LLN to detect link failure and attempt to fix the fault locally using less pessimistic strategies. The difference between NST, AODV and AOMDV is that the NST uses an optimistic recovery where no repair is required and the failure can be rectified by retransmitting the packet again. AOMDV is less optimistic than NST where the next-hop is declared as faulty and an alternative back-up route is used before the route is re-established locally. If the local RD is unsuccessful, the packet is dropped and a RERR packet is returned to the source. All variants of AODV use global RD as the last attempt

to rediscover a new route.

Table 2.3: Route detection and recovery mechanisms in different AODV routing protocol. The actions (in bold) highlight its main feature used to rectify failures.

| Reactive routing protocol | Failure Detection | Route Recovery | Usage |
|---|---|---|---|
| TinyAODV (Pessimistic) | MAC LLN | • **Global RD** | • Less routing overhead generated<br>• Handle permanent failure. |
| AODV (Less Pessimistic) | MAC LLN / *Hello* Packet | • **Local RD**<br>• Global RD) | • Suitable for large network<br>• Handle permanent node failure |
| AOMDV (Less Optimistic) | MAC LLN / *Hello* Packet | • **Backup multiple route**<br>• Local RD<br>• Global RD | • Suitable for small network (less than four neighbours).<br>• Handle single next-hop failure. |
| NST-AODV (Optimistic) | MAC LLN | • **Layer 3 ReTransmission (RT)**<br>• Local RD<br>• Global RD | • Handle sporadic transient failure . |

**Routing Selection Mechanism**

Routing metrics are used by the routing algorithm to decide which neighbouring node to become the next-hop node. Perkins and Royer (1999) propose the use of the number of hops as the metric to decide on the next node to forward the packet as the default metric in AODV. This metric is commonly used in most AODV routing protocol as it provides the shortest path to the sink. Boughanmi and Song (2008) have shown that the hop count metric can minimise the packet latency and the total number of transmissions required between the source and the sink. However, it does not take into account other operating factors such as the node's energy availability, the network utilisation and the radio signal quality. Alternative metrics have been proposed to improve the transmission rate. For instance, Chen et al. (2006) propose the use of LQI to determine the link quality and avoid lossy radio link and transient failure and Younis et al. (2002) propose energy awareness metric computed from the nodes to avoid node failure due to energy depletion. Although the proposed metrics may help to avoid a bad link, it

does not play a role in providing any assistance during the maintenance or repair of the route. For WSNs to respond an unexpected hardware or software failure, the node requires an online route management technique to detect and recover from failure.

#### Route Failure Detection

The route maintenance procedure is usually triggered locally when a node detects a link failure between itself and the next hop node. Link failure can be detected by: (i) the use of periodic *Hello* messages received from a neighbour (ii) the use of feedback known as Link Layer Notification (LLN) received from the link layer. In the former approach, *Hello* messages are broadcast by a local node periodically to inform the neighbouring nodes of its state. This process may congest the local network making it inaccessible. In contrast, LLN is provided by the link layer after a numbers of unsuccessful retransmissions (3 retransmissions). It does not congest the networks compared to the *Hello* messages. According to Chakeres and Belding-Royer (2005), the LLN is suitable for dynamic topologies.

## 2.9 Current Evaluation methodology

In a review conducted by Kurkowski et al. (2005) on 151 network research papers published in ACM Mobihoc from 2000-2005, 75.5% of the works have used network simulator to test their protocols. Only 60% has stated the number of iterations performed. 12.5% has shown statistical confidence in their results, with none of them addressed the randomness of the experiments. This can dramatically affect the confidence in the results obtained.

In this section, we have performed similar survey on 50 WSNs papers published in Mobile Adhoc and Sensor Systems (MASS) conference from 2010 to 2012, to investigate and update whether any attempt has been made to improve the confidence of the published work and satisfying the experimental properties in the area of WSNs, 5 years after the publication of the previous survey (Kurkowski et al., 2005). MASS is a leading conference in the WSNs community and 50 of the articles reviewed are related to experimental methods and results. Based on our survey, most of the findings in Kurkowski et al. (2005) still hold. We have found out that 80% of the published results are still based on simulation, 12% are hardware and 8% are mixed or hybrid approaches. However, 12 out of the 40 simulation papers did not state the type of simulation tools and 10 papers developed their own simulator but none of them provide their source code

making it difficult to reproduce their results. 60% have described their experimental setup and the use of pseudo-random generators to produce input data has increased significantly. However, there is still a lack of confidence on the simulation results published and the use of statistical analysis on the output results was found to be limited (Pawlikowski et al., 2002). Based on our current survey, only 17.5% have shown error bars in their graphical results with only 8% have stated the confidence interval (CI). None of the papers has performed any significance tests on their results and the issue of aleatory uncertainty was not addressed appropriately. Although 60% have repeated the experiment $n$ times, how $n$ is obtained statistically was not described. This can make it very difficult to ensure that the sufficiency of data to minimise uncertainty and their results are both statistical and scientific significant. Any significant results observed in an experiment, whether from simulation or hardware, must be at a specific *statistical confidence level*, *repeatable* and textitunbiased as any improper data analysis can often lead to incorrect or misleading results, threatening the credibility of the published work. The lack of scientific analysis and evaluation can affect the validity properties of the dependable system (Avizienis et al., 2004b). Contradictory result observed by Stetsko et al. (2011), in an experiment to evaluate the uncertainty between four different simulators, calibrated using data collected from real hardware experiments, has reiterated the need of achieving confidence in experiments using a more scientific approach. The summaries of the survey, based on the three criteria stated, is shown in Table 2.4.

Table 2.4: Survey from 50 papers published in IEEE MASS Conference 2010-2012 have shown that there is insufficient evidence to support and evaluate the dependability claims made in Kurkowski et al. (2005) due to lack of repeatability.

| **Evaluation Criteria** | **Percentage** | **Comments** |
|---|---|---|
| *Statistical Valid* | 17.5% | Shows error bar on plot |
| | 8% | States the confidence interval (CI) |
| | 0% | Performs significance tests |
| *Empirically Valid* | 66.7% | States the tools used |
| | 60% | Provides simulation parameters |
| | 0% | Provides source code |
| *Uncertainty Analysis* | 60% | Uses seeded inputs |
| | 60% | Performs repeated runs |

Another issue that can affect the credibility is the lack of various real world

scenario used in the experiments (Andel and Yasinsac, 2006). For example, protocol A may outperform B in environment X, but the opposite in environment Y. As shown in Chapter 3, The NST did not perform better than AODV when the failure duration increases from 0.5 to 20 seconds. NST was initially tested on real hardware by Gomez et al. (2006) to evaluate the energy efficiency and route change latency, but performance evaluation of NST was not conducted rigorously against other test scenario and no comparative performance is performed against other routing protocol. Hence, it is difficult to verify the reliability of NST.

### 2.9.1 The hybrid approaches

The survey in Section 2.9 has shown that 84.5% of the papers reviewed still rely on simulation tools to test and evaluate their new protocol as they allow significant levels of testing to be performed at reasonable cost. The simulation is usually repeatable and controllable. Unfortunately, the performance observed in simulation often differs considerably and does not perform as predicted during deployment (Langendoen et al., 2006). There is a *reality gap* between the simulation and real hardware experiment due to the simplified models, such as the radio and communication models, built in the simulator (Pham et al., 2007). Langendoen et al. (2006) has suggested the use of exhaustive testing to reduce errors between simulations and real deployment. It is not sufficient to use simulation alone as a validation tool. Instead, a mixed approach between simulation and hardware is more appropriate as simulated result can be cross-validated using small scale experiment on real hardware in a controlled environment.

### 2.9.2 Reality Gap in the Current Evaluation Approaches

There is still some mismatches between physical hardware and simulation results. Pham et al. (2007) attempt to validate the wireless channel model of Castalia by comparing the performance of MAC protocol. Although the simulator was calibrated using the connectivity maps obtained from the real experiment, there are discrepancies in Packet Reception Ratio (PRR) between simulation and the real hardware. WSN simulations usually use unrealistic assumptions, such as circular radio transmission area, and no fading or shadowing phenomena. These assumptions can lead to simulation results that differ significantly from experimental results. Shakshuki et al. (2009) implement the energy model taken from real sensor node to their java-based simulator and NS-2, in order to make the results between simulator and real world deployment closer. Their studies have shown that the results from a custom-made simulator did not perform better than

NS-2 because the overhearing and packet collision effect were not implemented in their simulator. They suggested that it is necessary to develop or use an existing network simulator that provides a complete protocols stack, such as NS-2, as their results have shown similar patterns to real. Bergamini et al. (2010) evaluate the performance of two routing protocols (NETSET and Gossip) using two different simulators (Castalia and NS-2) and validated the simulated results against the results obtained from 34 TMote Sky sensor nodes. They suggested that a simple parameter tuning may significantly increase the accuracy in the simulators. By tuning the simulator using the data captured from ZigBit-A2 nodes and applied them to Castalia, Gama et al. (2011) have shown that the simulated results obtained match satisfactory to those obtained in real conditions. However, from their results, it is not possible to show whether their results are both statistically and scientifically significantly better without performing any statistical hypothesis test.

## 2.10 Summary

In this chapter, the concept of dependability is discussed and the factors that can affect the dependability of WSNs are highlighted. Fault tolerance techniques have been investigated that looked into various detection, diagnosis and recovery approaches. Many algorithms have been proposed to assist in providing a fault-tolerant WSNs. However, some of the techniques may incur additional overheads and is based on specific static failure model that does not change during operation. With the complex and dynamic characteristics of WSNs, an alternative anomaly identification and recovery approach that can address these issues as well as adapt to the environmental changes is required. Having reviewed some of the existing detection, diagnosis and recovery in WSNs, it is crucial that the fault tolerance for WSNs is distributed, autonomous and lightweight in order to reduce energy consumption. This thesis will attempt to provide an optimal design for fault-tolerant WSNs based on the following key characteristics:

- ***Distributed architecture***: Detection should be done in a distributed manner to reduce communication overhead.

- ***Stateless***: Individual nodes should identify and recovery from the fault autonomously and independently in the local environment. There should not be any centralised control to make the protocols scalable and robust to failure.

- ***Lightweight:*** The algorithm must have a low computational complexity. It should also be integrated to reduce the processing overhead and recovery time.
- ***Unsupervised learning:*** Unsupervised, non-parametric learning is preferable as it is not easy to determine and obtain the normal predefined data.
- ***Adaptive***: It should be able to learn and dynamically adapt the detection and recovery model online. An adaptive model generated using a combination of different data vectors are required to capture the spatial and temporal variants of WSNs
- ***Self-healing:*** As the inaccessible nodes are difficult to maintain, the network should exhibit the self-healing properties to adapt to the dynamic network topology and to maintain the operation of the WSNs. One approach is to apply the bio-inspired algorithm to assist in network recovery using the current state of the environment.

CHAPTER

# THREE

# Evaluating the Fault Tolerance of the WSNs Routing Protocols

To investigate the hypothesis formulated in Section 1.2, this chapter presents an experiment to analyse the fault tolerance of the multihop network communication in WSNs using the current state of the art approaches applied in WSNs. The aim of the chapter is to produce an experimental framework that will be used throughout this thesis to evaluate the dependability of different WSNs routing protocols exposed with different failure characteristics. Based on the survey conducted in Section 2.9, Section 3.1 formulates the research methodology using the current state of the art approach in WSNs to evaluate the performance of the routing protocols. A systematic evaluation to investigate the fault tolerance of the routing protocols introduced in Section 2.8.1 is presented Section 3.2. We summarise the results observed from the experiments in Section 3.3.

## 3.1 Materials and Methods

In this section, we formalise the materials and methods that are exercised in this thesis based on the literature reviewed in Section 2.9 to evaluate the performance of the routing protocols. To ensure that the experiments performed in our works follow a systematic evaluation based on the current practises taken by WSNs research community, an experimental framework is formulated in Figure 3.1 that will be used to evaluate the routing protocol reliability in WSNs. The first step in any experiment usually involves the formulation of the objectives and any

assumptions made. This is followed by the description of the design of experiments. It is necessary to reduce the uncertainty introduced into the experiment by repeating the experiment sufficiently before the results are analysed visually and mathematically. The following subsections elaborate the activities involved in each of the stage.

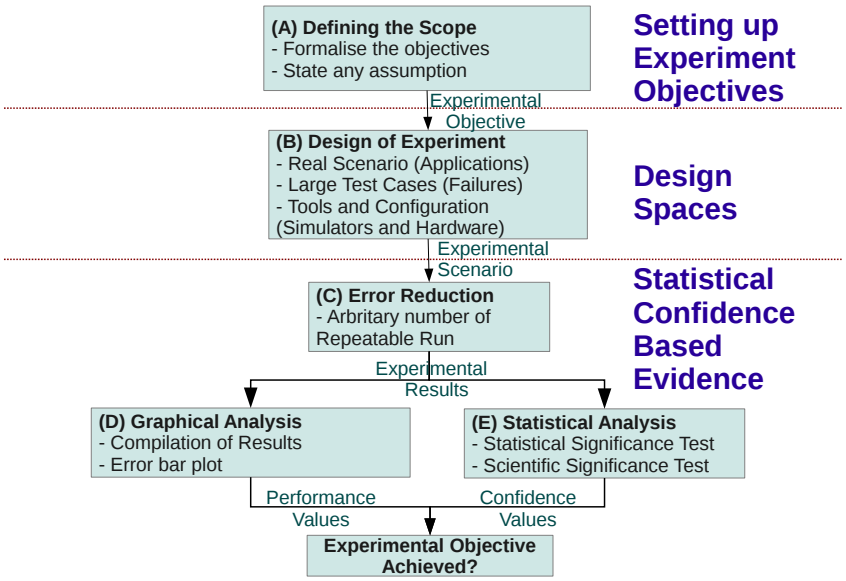



Figure 3.1: An Experimental Framework based on the existing State of the Art experimental techniques in WSNs

### 3.1.1 Application Environment and Topology

According to Akyildiz et al. (2002), WSNs can be deployed in a predefined location or random placed across an area of interest. Random deployment is usually applied when the area to be monitored is not accessible. A large number of nodes is usually deployed to ensure sufficient nodes are available to cover the sensing area and communication range. However, random deployment of WSNs in the real world, if any, is limited in the literature because it can be very expensive and difficult to determine the number of nodes required for sufficient coverage (Zou and Chakrabarty, 2007). The position of the nodes is usually pre-determined and placed carefully to ensure that the nodes are within their transmission range. Fixed deployment can be deployed outdoor or indoor. In outdoor deployments, a grid topology is usually used to optimise the node's placement. A number of nodes ranged from 16 to 900 nodes are placed in parallel ($n$ by $n$) (Hatler, 2012). In an indoor application, two or more nodes are usually placed in the room and along the corridors to provide redundancy (Chipara et al., 2010). Node redun-

dancy plays an important role to provide fault tolerance. In our studies, we follow the node redundancy approach in a grid-based topology similar to a vineyard monitoring proposed by Beckwith et al. (2004) to mimic a realistic deployment. A $n$ by $n$ predefined grid network with $N$ static nodes is proposed as shown in Figure 3.2 ($n$=7 and $N$ = 49). The values of $n$ and $N$ can be different in real world deployment depending on the sensing area and the application area.

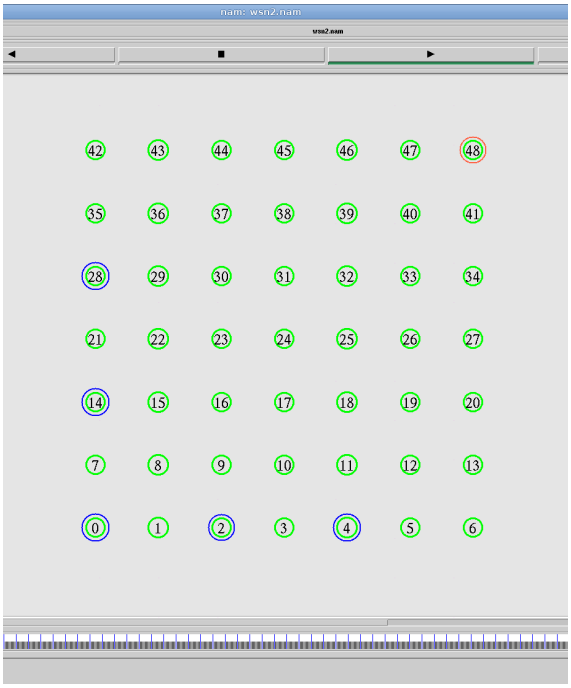



Figure 3.2: A 7 by 7 grid topology based on multipoint to point network to model an outdoor deployment where node 0, 2, 4, 14, and 28 (in blue) are evenly distributed to send periodic packets to sink (node 48, in red).

### Traffic Flow

WSNs are usually used for sense and send applications. The flow of data packets in WSNs depends on the requirements of the application running. It can be categorised into point-to-point, point-to-multipoint, multipoint-to-multipoint and multipoint-to-point (Akkaya and Younis, 2005). Point-to-point provides direct communication between the sender and receiver that can be triggered in two methods. Firstly, a source may send the measurement to the destination periodically or when an event occurs. Secondly, a destination node may prompt for measurements from a specific node. Point-to-multipoint traffic is usually generated by a broadcast or flooding algorithm (Zhang et al., 2012). A typical WSNs application is usually modelled by multipoint-to-point where information are generated from many nodes and forwarded to the sink or cluster head (Langendoen et al.,

2006; Liu et al., 2010). A multipoint-to-point becomes a multipoint-to-multipoint when there are more than one sinks. In our simulation, we simulate a multipoint-to-point network traffic flow as it is common for the sensed data to be collected in one location across multiple hops (Chipara et al., 2010). For example in Figure 3.2, nodes 0, 2, 4, 14, 18 are assigned to send data packets to the sink (node 48).

#### Traffic Pattern

Demirkol et al. (2006) categorise the WSNs traffic pattern into event-driven and periodic data-driven. In event driven, data packets are usually send to the user when an interesting phenomenon is detected by the sensor nodes. This traffic pattern is usually used for applications such as target detection and tracking (Lédeczi et al., 2005). For a monitoring application, a periodic data driven traffic model are usually applied. A constant bit rate (CBR) is often used where the data are usually transmitted by a node at a predefined interval (Cui et al., 2005). This CBR can be used to represent the sensed data from an environmental monitoring application or the node's health check information for event-based application. For example, safety critical applications such as fire detection and intruder detection usually perform maintenance check on the network to ensure that the nodes are always ready to sense and forward data to the sink when fire occurs (Wang and Kuo, 2003; Wu et al., 2010a). The nodes usually transmit periodical heartbeat packets regularly across the network to check the sink is still reachable. Hence, we apply the CBR traffic to flow from the sources (node 0, 2, 4, 14, 18) every $t$ seconds starting at different time intervals $i$ to reduce the probability of packet collisions.

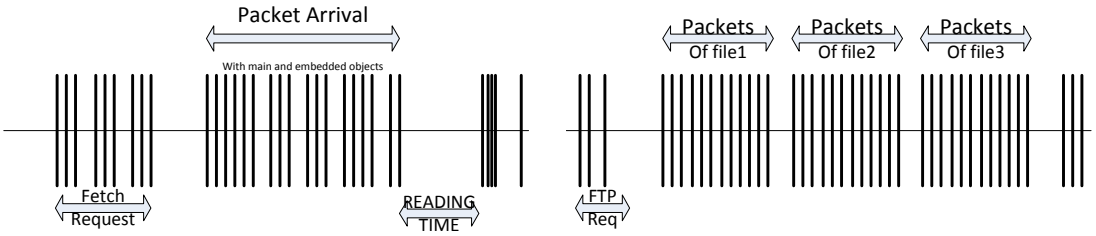
A set of baseline simulations using different $t$ and $i$ is run to determine the values of $t$ and $i$ to ensure that the routing protocols can produce significant results when failures are introduced. Setting a low $t$ value may not affect the routing behaviours and the results. From the baseline simulation, $t$ is set to 0.5s to allow the failing node to disrupt the sending of the packet. $i$ is set to 0.1s to reduce failures due to collision.

### 3.1.2 Failure Model

Errors on the radio communication can occur in WSNs as the open access radio channel is shared by other devices. The interference in operating environment can affect the routing protocol operating in the node. Different failure models have been proposed and applied in WSNs. Most of these works concentrate on fault models to detect intentional error. Unintentional error is common in the

communication between the two nodes and may degrade the network performance. As the 2.4-GHz ISM band occupied by WSNs is shared by other home devices such as Wireless Network, microwave oven (MO) and cordless phones, the communication between the nodes can be disrupted when the signal strength emitted by these devices are higher than the transmitting node's signal strength (Reis et al., 2006; Mahanti et al., 2010). Mahanti et al. (2010) report that some interferences are relatively short, but some have a long duration. For example, the average active period (in min:sec) for microwave ovens (2:12), and digital cordless phones (7:32). Hence, it is important to be able to recognise and distinguish the interference patterns in order to construct appropriate test cases into any evaluation.

3GPP2 (2009) and Zhuang et al. (2008) highlight that a typical wireless application session can be observed as a sequence of distinct ON and OFF period as shown in Figure 3.3. This ON and OFF patterns have been used in software-based traffic generators to generate WLAN traffics (Botta et al., 2010). Hence, in our research work, we apply the ON/OFF model to capture the interference phenomenon generated by WLAN traffic.



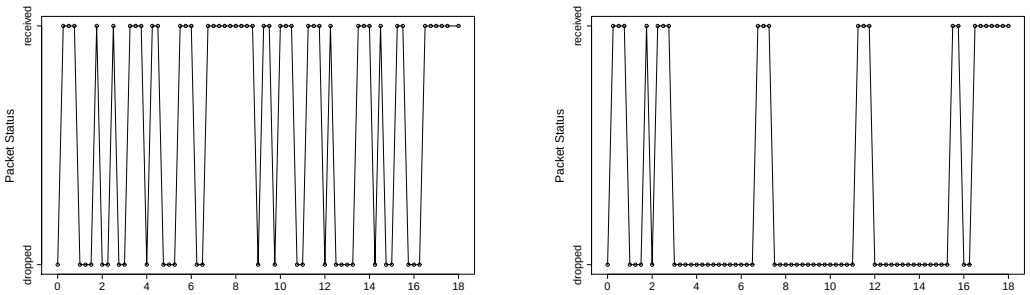

(a) Non-uniform arrival of HTTP packets.

(b) The arrival of FTP packets is equally separated.

Figure 3.3: A typical WLAN traffic patterns for web browsing (http) and File Transfer (FTP)

In our ON/OFF interference model, the ON period will represent the failure duration ($f_d(t)$) (The interfering devices is ON). The OFF period will represent the interval ($f_i(t)$) between the occurrences of interferences (The interfering device is OFF). The ON period can be classified into transient failure and permanent failure. In permanent failure, the node is unable to communicate with any node permanently ($f_d(t) = \infty$). It is usually created by obstacles such as metallic objects that reduce the radio range of the sensor node. Hence, the interference is always ON when it occurs. For transient failure, it has an irregular and unpredictable duration that differs between activities. These failures may gradually reduce or generate irregular packet delivery rate.

To investigate the effect of interference on the packet delivery rate, we perform an experiment to capture the number of packets successfully received during radio interference using a TelosB mote (Polastre et al., 2005). A sensor node is configured to transmit a continuous fast stream of packets (250ms per packet) without CCA and acknowledgement. Due to the limitation of the clock in TelosB, the maximum transmission rate is equal 250ms. The number of packets send successfully during WLAN interference is collected to observe the effect of interference on the transmission. We expose the WSNs to two different WLAN traffics (FTP and HTTP traffics) each with different pattern as shown in Figure 3.3. From the results, the patterns of the packet transmission are shown in Figure 3.4.




(a) Packet received during www interference

(b) Packet received during ftp interference

Figure 3.4: The patterns showing the number of successful packets received by a node during interference. The maximum duration for FTP traffic is 3.5s while the minimum duration for www traffic is 0.25s (1 packet).

The failing pattern observed in Figure 3.4 is similar to Figure 3.3. Using that pattern, it is possible to produce a model of the failure pattern using with the ON and OFF model. Using the results observed, the interference model is classified into short, medium and long duration. and $f_d(t)$ and $f_i(t)$ can be set accordingly as shown in Table 3.1.

Table 3.1: The values of ON/OFF periods representing short, medium and long interferences.

| Interference | ON period | OFF period |
|---|---|---|
| Short | $f_d(t) < 0.25s$ | $f_i(t) < 0.25s$ |
| Medium | $0.25s < f_d(t) < 4s$ | $f_i(t) < 0.25s$ |
| Long | $f_d(t) > 4s$ | $f_i(t) < 0.25s$ |

To simulate the interference failure in our simulations, an active node along the route is shut down during the ON period ($f_d(t)$) and turn on during OFF period ($f_i(t)$). Although $f_d(t)$ and $f_i(t)$ may be context dependent and the ex-

act channel usage of the WLAN devices in the real world may differ from that of the particular applications studied in our controlled experiments, the general characteristics observed should be similar.

### 3.1.3 Experimental Tools

To avoid any failures during operation, a newly design network protocol needs to be tested thoroughly before any deployment in order to eliminate any errors induced during design or implementation. The three commonly used techniques to test and analyse the operation and performance of wired and wireless networks can be categorised into analytical, computer simulation, and physical deployment. Due to the complexity of WSNs, such as dynamic deployment environment, unpredictable behaviour, and limited energy utilisation, it is complicated and unfeasible to model and analyse the complex WSNs using traditional network analytical methods (Hawrylak et al., 2009). The use of oversimplified assumptions can produce results with limited confidence. Real physical hardware deployment is usually expensive and time consuming. It requires a huge effort to deploy especially in a large scale WSNs. Therefore, simulation is usually preferable to evaluate the WSNs as large amounts of simulated data can be collected cheaply and quickly for analysis.

#### Simulation Software

Different WSNs simulation tools have been developed. Dwivedi and Vyas (2011) list fourteen WSNs simulators commonly used in WSNs experiments to evaluate different aspects of WSNs such as the application, the network algorithm, and the node behaviours. The number of simulators keeps on increasing as a new simulator is proposed to tailor for different applications and scenarios. NS-2 is the most widely used simulator to evaluate and analysed the network behaviour in WSNs prior to deployment (Dwivedi and Vyas, 2011). It is initially developed in 1989 as a traditional network simulator. Since then, NS2 has gained popularity among the WSNs community compared to other simulators. Various enhancements have been implemented in NS2 including adding the IEEE (2006a) 802.15.4 MAC and PHY layer to improve the validity of the simulation (Zheng and Lee, 2006). NS-2 contains a large numbers of external protocols and modules required including routing protocol, battery models, and scenario generation tools. Each simulation can be automated and repeated until a sufficient level of confidence in the results has been achieved. The simulations are run at the packet level that can generate detailed logs required for our studies.

In this research work, we will be using the NS-2.34 simulator to evaluate the routing protocols in our research work as it is an open-source event-driven simulator commonly used in the WSNs community (NS2, 2002). It can be extended to support new protocols and can integrate with external script to support new function. Although the large amount of logs generated by NS2 can be difficult to analyse, we address this issue by post-processing the logs using automated scripts written in Perl and shell scripts [1] that allowed us to compute the performance metrics required for the analysis.

### 3.1.4 Simulation Methodology

In order to get a good estimate for the steady-state packet drop rate, the simulation has to be run for a period of time. Furthermore, the transient phase $Tr(t)$ at the start of the simulation has to be removed and $Tr(t)$ depends on the size of the network. A preliminary baseline simulation on the network size of $N$=49 nodes (7 by 7) shows that a simulation time of 120 seconds and a $Tr(t) = 10s$ are needed to initialise the network. To reduce the errors in the simulation, we repeat the experiments $n$ times and arbitrary set $n$=50. We will later show in Chapter 6 that how the number of runs to reach a level of confidence in the results generated can be determined systematically.

There are three radio propagation models available in NS-2.34 namely free space model, two-ray ground reflection model and the shadowing model (NS2, 2002). The two-ray ground reflection model is commonly applied in the NS-2.34 simulation. Although the log-normal shadowing path loss model provides a more realistic propagation to model fading effect in the noisy environment, the two parameters, namely, the path loss exponent and the log-normal shadow variance have to be measured and configured accordingly (Zamalloa and Krishnamachari, 2007). The nodes near to the boundary of the transmission range can only probabilistically communicate when simulating using shadowing model. As a result, random packets are dropped. This can affect the observed results between simulations. With a two-ray ground model, the communication failure can be managed and controlled. Hence, the two-ray ground propagation model with CSMA/CA protocol are used in our simulation. The IEEE 802.15.4 physical and MAC protocols implemented by Zheng and Lee (2006) in NS-2.34 are also enforced to ensure that our simulation follows the 802.15.4 standard.

In summary, the parameters to be used in our work are provided in Table 3.2.

[1] Downloadable at http://rtslab.wikispaces.com/file/view/ns2script.tar

Table 3.2: Generic NS-2 Parameters used in simulations

| **Parameters** | **Values** |
|---|---|
| *Simulation area:* | 200x200m |
| *Number of nodes* | 49 nodes |
| *Transmission interval:* | 0.5 s |
| *Propagation model:* | Two-Ray Ground |
| *MAC*: | 802.15.4 (CSMA/CA) |
| *Routing Protocol:* | AODV, NST, AOMDV, TINYAODV |

### 3.1.5 Evaluation Metrics

Reliability and latency are the two of the most important metrics in safety critical system. They must therefore be considered during network design. Furthermore, energy consumption remains a primary design concern in WSNs as a reasonably long network lifetime is still necessary. According to Suriyachai et al. (2012), the performance of a WSNs protocol can be measured based on the time domain or reliability domain. The performance in the time domain relies on when data is received at the destination and its ability to meet a specific deadline. Time domain performance can be represented by packet delay. In terms of reliability domain, it relies on how much data is received at the destination. Delivery ratio and packet loss rate are measurements often used to represent this reliability performance. To compare the performance between each routing protocol required for the studies, the following performance metrics are employed to test the objective defined in Section 2.8:

- **Packet Delivery Rate:** Packet Delivery Rate (PDR) is the ratio of total number of data packets received, $P_r$ to total number of data packets sent $P_s$. PDR is commonly used to measure the reliability of the network. It allows us to compare and measure how many packets have been successfully delivered by the routing protocols.

$$PDR = \frac{\sum P_r}{\sum P_s} \times 100$$

- **Average Energy Consumption:** The average energy consumption ($E_{consumed}$) calculates the average amount of energy being utilised in the node to achieve the PDR. It allows us to evaluate the amount of energy required for routing. The lower the $E_{ave}$, the better the routing algorithm is as it has increased the lifespan of the network. Hence, routing protocol, that has the lowest $E_{ave}$,

is preferable.

$$
\begin{aligned}
E_{ave} &= \frac{\sum_{i=1...n}^{N} E_i(t_0) - E_i(t_x)}{N} \\
E_{consumed}\ (in\ \%) &= \frac{E_{ave}}{E_i(t_0)} \times 100 \\
where\ N &= Total\ number\ of\ nodes \\
E_i(t_0) &= Initial\ Energy\ in\ the\ node \\
E_i(t_x) &= Energy\ Remains\ in\ the\ node
\end{aligned}
$$

- **End to End Delay:** End to end delay ($Delay_{ave}$) is the sum of the delays of each packet received over N, the total number of packets received. It indicates the effectiveness and availability of the protocol and compares the network latency between different protocols.

$$
\begin{aligned}
Delay_{ave} &= \frac{\sum_{p=1}^{N} t_{rec}(p) - t_{sent}(p)}{N} \\
where\ N &= Total\ number\ of\ packet\ received \\
t(p_{rec}) &= Time\ Packet\ Sent\ at\ source \\
t(p_{sent}) &= Time\ Packet\ Received\ at\ sink
\end{aligned}
$$

- **Normalised Routing Overhead:** Routing overhead is calculated as the normalised ratio of total routing packet transmitted to the total data packet received. It is an important metric used to analyse the routing protocols performance as it measures the routing overhead generated to deliver a data packet. A low normalised routing overhead value is desirable as it indicates small amount of energy is wasted during RD.

$$
Route_{overhead} = \frac{\sum Number\ of\ RREQ}{\sum Total\ Data\ Packet\ Received}
$$

### 3.1.6 Statistical tools

Data generated from multiple measurements can be subject to error. It is necessary to reduce and understand the error by applying statistical analysis to summarise those observations and quantifies the uncertainty in the measured vari-

able. In our work, we will use the R statistical tool [2] to explore data sets, analyse the statistics and plot the graphical representation of data results (Ihaka and Gentleman, 1996). The use of statistical tools to analyse results collected from WSNs experiments are limited. It is necessary to analyse the significance and confidence of the results obtained to show that the performance improvement observed is significant and different. Statistical tests allow us to analyse the probability that the results are different and whether the difference is significant. Two non-parametric statistical tools can be applied to test the difference namely statistical and scientifically significance test. Statistical significance test estimates the probability that result if the results obtained have come from the same distribution. However, the test does not show how much the difference was. By performing scientific significance test, the differences between the two distributions can be estimated. Although non-parametric tests are weaker and are less likely to reject the null hypothesis when it is false, this problem can be addressed by using a large sample size to perform the test. According to Conover (1999), a nonparametric test will require a slightly larger sample size to have the same power as the parametric test. Hence, we repeat the experiments $n$ times to generate a large sample. We will later show in Chapter 6 how to determine $n$ systematically.

**Statistical significance test**

Statistical significance test can be used to determine whether the difference in performance observed in the results is likely to have occurred due to random chance with the samples available, i.e. whether protocol X is really better than Y or whether the results are so close than differences are purely random. In order to determine and compare the relationship between the two samples collected from the experiments, Mann-Whitney-Wilcoxon test, also known as rank-sum test, is applied to compute the $p$-value (Wilcoxon, 1945). This non-parametric test is used as it does not make any particular assumptions about the distribution of the result, avoiding the need to verify the data conform to the test assumption. Although a normality or other similar distribution test can be performed to determine whether the data set exhibit a specific distribution, various reports have shown that the data generated by the WSNs cannot be modelled by a known distribution. To show that our results generated from the WSNs simulation are not normal, the Shapiro-Wilk normality test is applied to perform the hypothesis test to reject null hypothesis (Shapiro and Wilk, 1965). The null hypothesis states that the results are normally distributed. The test results for samples collected from

[2]The scripts downloadable at http://rtslab.wikispaces.com/file/view/plotgraph.tar

one of the simulated scenario have rejected the null hypothesis as the $p$-values $<< 0.05$ (Table 3.3).

Table 3.3: Results from Shapiro-Wilk Test show that the distribution of the results generated from a WSN simulation does not follow a normal distribution ($p$-values $<< 0.05$).

| ENG | PDR | RT | DLY |
|---|---|---|---|
| 0.4936 | 1.47E-09 | 9.41E-11 | 6.07E-13 |
| 0.5996 | 3.71E-10 | 3.19E-08 | 4.49E-12 |
| 0.0271 | 5.21E-08 | 1.31E-07 | 4.24E-09 |
| 0.7244 | 3.91E-07 | 0.0018 | 1.64E-11 |
| 0.0022 | 9.33E-07 | 0.0017 | 1.58E-10 |
| 0.1651 | 5.31E-06 | 1.14E-02 | 8.24E-09 |
| 0.0478 | 6.39E-06 | 9.31E-09 | 3.23E-08 |
| 0.0120 | 1.29E-06 | 2.14E-07 | 6.96E-08 |
| 0.0006 | 1.93E-06 | 4.34E-08 | 1.99E-07 |
| 0.0027 | 6.74E-06 | 3.24E-08 | 9.90E-08 |
| 0.0018 | 2.75E-05 | 4.32E-07 | 1.70E-08 |
| 0.0297 | 4.82E-05 | 1.55E-05 | 1.94E-07 |
| 0.0023 | 0.0030 | 0.0002 | 8.68E-07 |
| 0.0018 | 0.0031 | 6.74E-06 | 2.68E-06 |
| 0.0089 | 0.0018 | 0.0004 | 6.43E-07 |
| 0.0014 | 0.0046 | 0.0005 | 7.09E-08 |
| 0.0007 | 0.0033 | 0.0014 | 4.68E-07 |
| 0.0023 | 0.0423 | 0.0012 | 4.02E-06 |

Another benefit of using the rank-sum test is the statistics generated from this test can be used to perform scientific significance test. Based on a pre-determined confidence level of X%, the results are shown to be statistically significant if the $p$-value $<= \alpha$. An $\alpha$ value of 0.05 is typically used, corresponding to 95% confidence levels (Wilcoxon, 1945).

A two-tailed test is applied in our analysis as the directional prediction of our data sample is unknown, that is whether the median of protocol A is better than protocol B is unknown. We are only interested in determining the different between the samples for this test. A two-tailed test is more rigorous and demands more evidence to reject the hypothesis. The use of one-tailed test alone to demonstrate significance is not sufficient (Gravetter, 2012). Hence, we will use a two-tailed test and apply the scientific significance test to help us to deduce the

direction of the relationship of the data (if any) by computing the effect size.

#### Scientific significance test

It is possible for the observed performance improvement between the protocols to be statistically significant due to the large amounts of sample used to test the protocol but the differences are small. It is important to examine the scientific significance of results to measure the difference in the data distribution or the effect size between the protocols. Another non-parametric test known as the Vargha-Delaney $A$-statistic is used to measure the effect size (Vargha and Delaney, 2000). $A$-value in the range $[0, 1]$ is obtained using the parameters collected from the previous rank-sum test. Using the guidelines proposed by Vargha and Delaney (2000), the range of $A$-values representing different effect sizes are presented in Table 3.4.

Table 3.4: The range of $A$-values proposed by Vargha and Delaney (2000) to represent different effect sizes.

| Large Effect Size | Medium Effect Size | Small Effect Size |
|---|---|---|
| $A$-value $<= 0.27$ | $0.27 < A$-value $<= 0.36$ | $0.36 < A$-value $< 0.44$ |
| $A$-value $>= 0.73$ | $0.64 < A$-value $<= 0.73$ | $0.56 < A$-value $< 0.64$ |

## 3.2 Performance Analysis of Routing Protocols

In order to analyse the competency of a routing protocol to tolerate failures, a set of objectives is defined as:

- To investigate the ability of a routing protocol to tolerate failures with different duration and density.
- To compare the performance of the AODV, NST, AOMDV and TinyAODV routing protocols injected with different failures.
- To identify the properties of a routing protocol used to rectify failures.

The results collected from the experiments are presented to investigate and evaluate the performance of the routing protocol operating in the nodes.

### 3.2.1 Observations

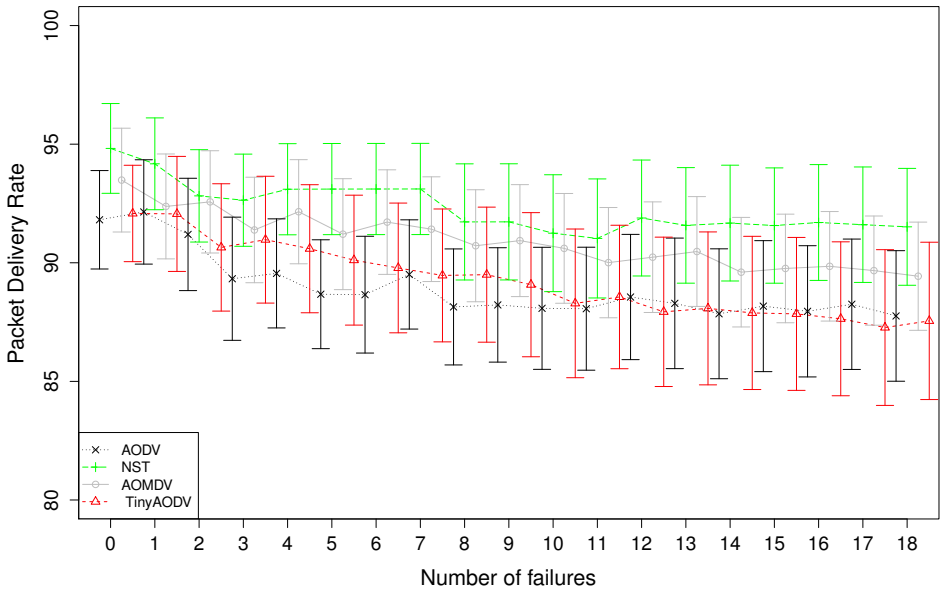
Figure 3.5 shows the PDR generated by each of the routing protocols under different number of failing nodes with transient (0.1s) and long (10s) failure duration. During the error-free condition (No fault is injected), NST (95%) can deliver more packets than AODV (92.4%), AOMDV (94%) and TinyAODV (92.5%). As the number of nodes injected with 0.1s failure duration increases, the number of packets delivered for all the routing protocols decrease gradually. In order to validate the results, a Mann-Whitney-Wilcoxon test is applied to check if the observed differences are statistically significant (Wilcoxon, 1945). Table 3.5 and 3.7 show that NST has a significantly higher PDR than AODV (Rank-Sum $p$-values$<$ 0.09 , Vargha-Delaney $A$-value$<$0.662 (Medium Effect Size)).

When a longer error of 10s is injected, the PDR observed in AOMDV lower than AODV, NST and TinyAODV. When more than 14 failing nodes, the PDR for AOMDV has dropped by half as shown in Figure 3.5(b). Although the means of the PDR of NST taken from 50 runs in Figure 3.5(b) is showing higher means than NST and Tiny-AODV, the statistical tests have revealed than the differences in PDR does not produce scientific values. For instance, the Vargha-Delaney test in Table 3.7 have shown that the differences are always not significant when there are more than 11 nodes failing with $A$-value $< 0.65$.

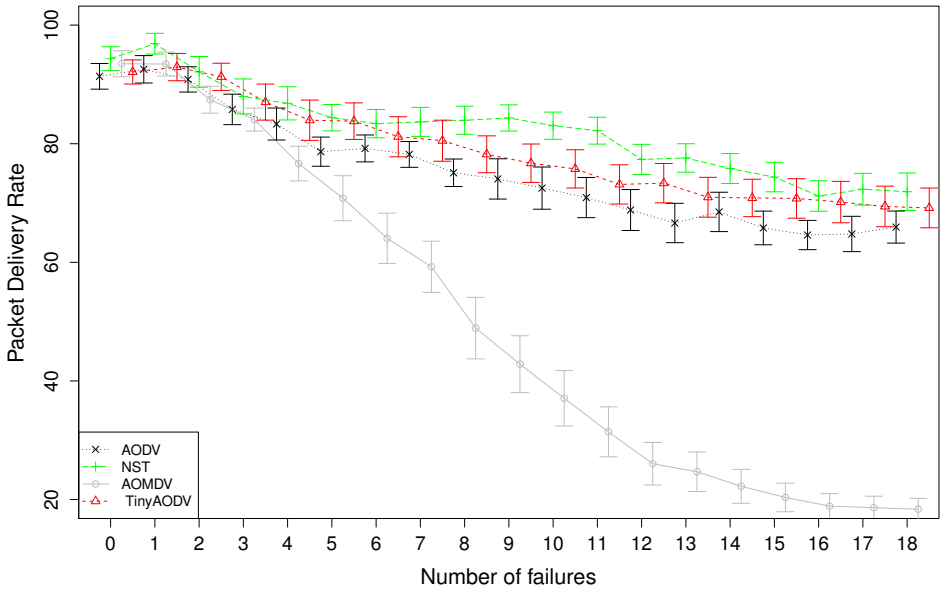
Figure 3.6 shows the percentage of energy consumption required for each of the routing protocols with increasing number of failures. Figure 3.6(a) shows that no significant difference in energy consumption between the AODV, NST and Tiny-AODV when more than two nodes are failed at the same time at random interval. When a 0.1s failure duration is introduced to one node in the network, the energy consumption in NST is 0.25% lower than AODV (rank-sum test, $p$-value = 0.044) and TinyAODV (rank-sum test, $p$-value = 0.022) as depicted in Figure 3.6(a). Although the $p$-values at 0.1s (1 failure node) are $< 0.05$, the differences between NST and AODV ($A$-value = 0.619), and TinyAODV ($A$-value = 0.634) exhibit small effects size. Hence, the differences is not significant between the three protocols. The energy consumption slowly increases as more failures occur as shown in Figure 3.6(a). AOMDV exhibits the highest energy consumption ($> 10\%$) compare to AODV, NST and TinyAODV ($< 6\%$). We believe this additional energy consumption is due to the *Hello* packets in the MAC layers.

Further assessment of the simulation log reveals that the total number of MAC packets transmitted and received in AOMDV is 42% more than NST-AODV as shown in Table 3.13. Hence, the use of *Hello* packets is not suitable for WSNs as it consumes more energy than LLN in AODV, NST and TinyAODV.

When the failure duration is increased to 10s, the energy consumed by AODV




(a) Packet Delivery Rate for failure duration 0.1s




(b) Packet Delivery Rate for failure duration 10s

Figure 3.5: The means computed from 50 runs have shown that the PDR of NST is higher than AODV, AOMDV and TinyAODV as the number of failing nodes increases.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **2.46E-006** | **0.0008** | **3.02E-006** | 0.0578 | 0.8902 | **0.0290** |
| 1 | **6.14E-005** | **0.0036** | **0.0003** | 0.0566 | 0.6030 | 0.1580 |
| 2 | **0.0093** | 0.0527 | **0.0290** | 0.1774 | 0.7615 | 0.4946 |
| 3 | **0.0051** | **0.0282** | 0.0749 | 0.2761 | 0.3552 | 0.8146 |
| 4 | **0.0027** | **0.0112** | **0.0299** | 0.5216 | 0.5227 | 0.6341 |
| 5 | **0.0008** | **0.0026** | **0.0205** | 0.7079 | 0.3572 | 0.3106 |
| 6 | **0.0010** | **0.0033** | **0.0168** | 0.9860 | 0.4724 | 0.2609 |
| 7 | **0.0023** | **0.0029** | **0.0118** | 0.5217 | 0.7072 | 0.1418 |
| 8 | **0.0019** | **0.0054** | **0.0394** | 0.7341 | 0.3460 | 0.0607 |
| 9 | **0.0016** | **0.0050** | **0.0349** | 0.7447 | 0.3441 | 0.0810 |
| 10 | **0.0032** | **0.0105** | **0.0314** | 0.6265 | 0.5134 | 0.1036 |
| 11 | **0.0037** | **0.0063** | **0.0304** | 0.3682 | 0.5087 | 0.0626 |
| 12 | **0.0010** | **0.0020** | **0.0100** | 0.3050 | 0.4972 | 0.0835 |
| 13 | **0.0070** | **0.0088** | **0.0494** | 0.2380 | 0.4835 | 0.0552 |
| 14 | **0.0031** | **0.0033** | **0.0527** | 0.2762 | 0.3260 | **0.0298** |
| 15 | **0.0066** | **0.0079** | **0.0498** | 0.1988 | 0.5087 | **0.0379** |
| 16 | **0.0028** | **0.0044** | **0.0297** | 0.2463 | 0.4188 | **0.0479** |
| 17 | **0.0077** | **0.0049** | **0.0326** | 0.1482 | 0.6005 | 0.0597 |
| 18 | **0.0039** | **0.0036** | **0.0353** | 0.2139 | 0.4483 | **0.0283** |

Table 3.5: *p*-values computed from Rank-Sum test for different PDR with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **6.52e-05** | **0.00778** | **9.59e-05** | **0.03421** | 0.69726 | **0.03218** |
| 1 | **2.68e-05** | **1.05e-06** | **4.47e-06** | 0.77037 | 0.83922 | 0.94692 |
| 2 | 0.09241 | **2.73e-05** | 0.11763 | **0.00293** | 0.80650 | **0.00013** |
| 3 | 0.18466 | **0.00242** | 0.70181 | 0.16846 | 0.40207 | **0.00309** |
| 4 | 0.21535 | **2.96e-05** | 0.81759 | **0.02139** | 0.48603 | **0.00396** |
| 5 | **0.02606** | **2.05e-05** | 0.28597 | **0.04111** | **0.03284** | **0.00071** |
| 6 | **0.04898** | **2.03e-06** | 0.62116 | **0.00017** | 0.11596 | **1.58e-05** |
| 7 | **0.01576** | **4.31e-08** | 0.91198 | **4.09e-06** | 0.10750 | **2.91e-07** |
| 8 | **0.00154** | **2.06e-10** | 0.16926 | **1.36e-06** | 0.15937 | **1.22e-08** |
| 9 | **0.00506** | **5.31e-13** | **0.03707** | **3.03e-08** | 0.49497 | **5.75e-11** |
| 10 | **0.00908** | **5.66e-13** | **0.04204** | **8.22e-10** | 0.46859 | **1.82e-12** |
| 11 | **0.00306** | **2.38e-13** | **0.01489** | **1.15e-10** | 0.61126 | **2.75e-13** |
| 12 | **0.03982** | **1.79e-13** | 0.33008 | **6.55e-12** | 0.25869 | **2.76e-15** |
| 13 | **0.00862** | **7.53e-14** | 0.09067 | **1.83e-12** | 0.21234 | **2.46e-15** |
| 14 | 0.08949 | **8.09e-14** | 0.17687 | **5.26e-13** | 0.57226 | **1.56e-16** |
| 15 | **0.00930** | **7.00e-14** | 0.43058 | **4.39e-14** | 0.12612 | **7.28e-17** |
| 16 | **0.03519** | **3.13e-14** | 0.98433 | **3.13e-14** | 0.06155 | **5.41e-17** |
| 17 | **0.03463** | **2.91e-14** | 0.67853 | **4.15e-14** | 0.13104 | **4.81e-17** |
| 18 | 0.11469 | **1.77e-14** | 0.78412 | **2.19e-14** | 0.17685 | **3.16e-17** |

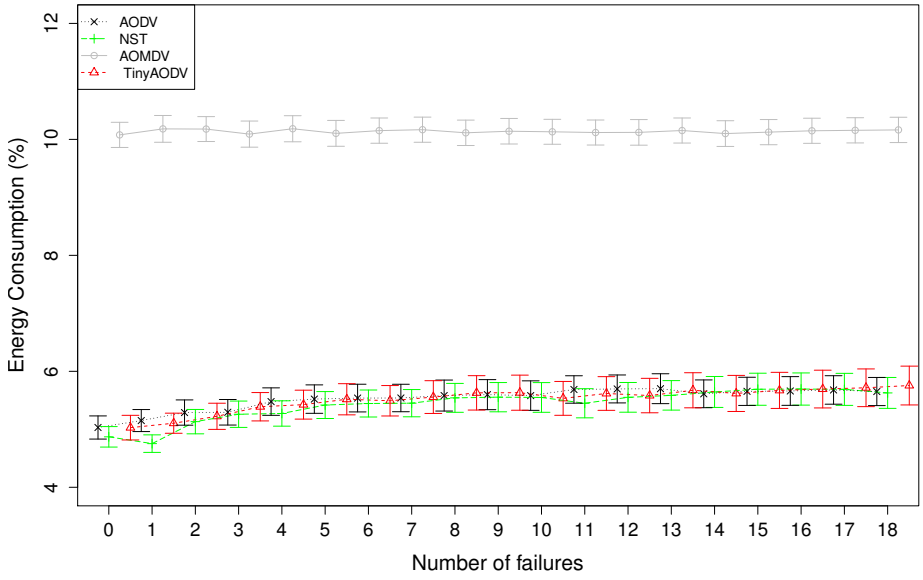
Table 3.6: *p*-values computed from Rank-Sum test for different PDR with 10s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **0.768** | *0.688* | **0.766** | 0.390 | 0.508 | 0.626 |
| 1 | **0.734** | *0.666* | *0.709* | 0.389 | 0.469 | 0.582 |
| 2 | *0.662* | 0.612 | 0.627 | 0.421 | 0.482 | 0.540 |
| 3 | *0.664* | 0.627 | 0.604 | 0.436 | 0.445 | 0.486 |
| 4 | *0.675* | 0.646 | 0.626 | 0.462 | 0.462 | 0.472 |
| 5 | *0.696* | **0.674** | 0.635 | 0.478 | 0.446 | 0.441 |
| 6 | *0.692* | **0.670** | 0.639 | 0.501 | 0.457 | 0.435 |
| 7 | *0.678* | **0.672** | 0.647 | 0.538 | 0.478 | 0.415 |
| 8 | *0.682* | **0.661** | 0.620 | 0.520 | 0.444 | 0.391 |
| 9 | *0.685* | **0.662** | 0.623 | 0.519 | 0.444 | 0.399 |
| 10 | *0.673* | 0.648 | 0.625 | 0.529 | 0.461 | 0.405 |
| 11 | *0.670* | **0.662** | 0.626 | 0.553 | 0.461 | 0.392 |
| 12 | *0.692* | *0.678* | 0.650 | 0.560 | 0.460 | 0.399 |
| 13 | *0.662* | **0.662** | 0.614 | 0.569 | 0.459 | 0.389 |
| 14 | *0.674* | *0.669* | 0.613 | 0.564 | 0.442 | 0.374 |
| 15 | *0.662* | *0.662* | 0.614 | 0.575 | 0.461 | 0.379 |
| 16 | *0.676* | *0.664* | 0.627 | 0.568 | 0.452 | 0.384 |
| 17 | *0.662* | *0.662* | 0.624 | 0.585 | 0.469 | 0.391 |
| 18 | *0.669* | *0.668* | 0.623 | 0.573 | 0.455 | 0.373 |

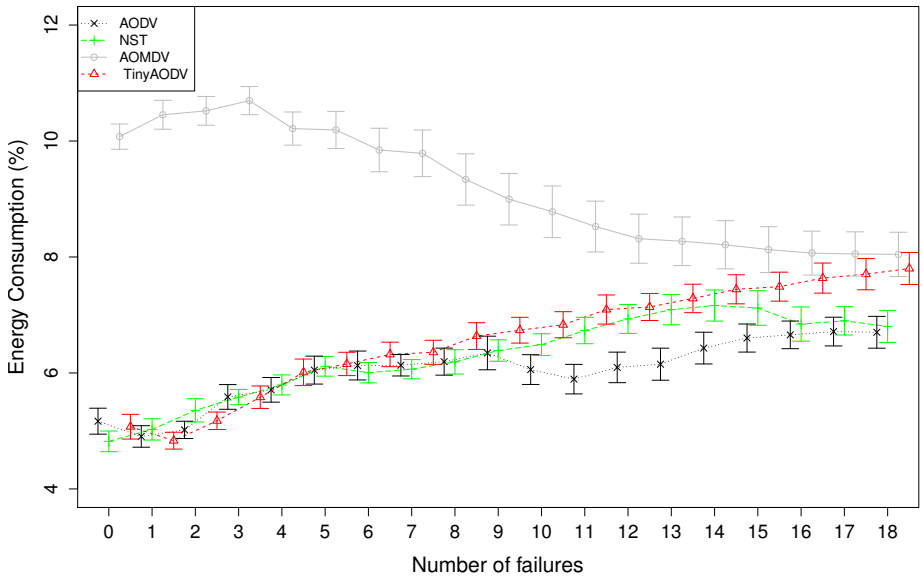
Table 3.7: *A*-values computed from Vargha-Delaney Test for different PDR with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows large effect size and in italic shows medium effect size.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **0.773** | *0.667* | **0.746** | 0.365 | 0.475 | 0.624 |
| 1 | **0.786** | **0.807** | **0.791** | 0.519 | 0.513 | 0.496 |
| 2 | 0.618 | **0.768** | 0.601 | *0.691* | 0.484 | *0.276* |
| 3 | 0.593 | *0.694* | 0.525 | 0.589 | 0.445 | *0.327* |
| 4 | 0.588 | **0.769** | 0.515 | 0.648 | 0.454 | *0.332* |
| 5 | *0.660* | **0.777** | 0.430 | 0.633 | 0.360 | *0.302* |
| 6 | 0.639 | **0.806** | 0.468 | **0.742** | 0.397 | **0.248** |
| 7 | *0.672* | **0.856** | 0.492 | **0.796** | 0.395 | **0.201** |
| 8 | **0.725** | **0.913** | 0.591 | **0.811** | 0.408 | **0.167** |
| 9 | *0.698* | **0.964** | 0.636 | **0.856** | 0.455 | **0.118** |
| 10 | *0.686* | **0.968** | 0.634 | **0.895** | 0.453 | **0.089** |
| 11 | *0.714* | **0.985** | *0.663* | **0.915** | 0.467 | **0.074** |
| 12 | 0.649 | **0.987** | 0.566 | **0.942** | 0.426 | **0.039** |
| 13 | *0.690* | 0.995 | 0.614 | **0.953** | 0.419 | **0.038** |
| 14 | 0.625 | **0.994** | 0.591 | **0.973** | 0.462 | **0.018** |
| 15 | *0.688* | 0.996 | 0.553 | **0.986** | 0.400 | **0.013** |
| 16 | 0.652 | **0.998** | 0.498 | **0.993** | 0.377 | **0.011** |
| 17 | 0.653 | **0.998** | 0.528 | **0.990** | 0.400 | **0.008** |
| 18 | 0.613 | **0.998** | 0.518 | **0.996** | 0.411 | **0.007** |

Table 3.8: *A*-values computed from Vargha-Delaney Test for different PDR with 10s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows large effect size and in italic shows medium effect size.




(a) Energy Consumption for failure duration 0.1s




(b) Energy Consumption for failure duration 10s

Figure 3.6: Lower energy consumption ($4.75\%$ with $5\%$ confidence level) due to retransmission is observed in NST-AODV when 1 node fails for 0.1s duration.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.691 | **4.70E-018** | 0.676 | **9.31E-018** | 0.989 | **9.87E-018** |
| 1 | **0.044** | **7.17E-018** | **0.022** | **1.44E-017** | 0.891 | **8.59E-018** |
| 2 | 0.350 | **8.27E-018** | 0.370 | **1.73E-017** | 0.820 | **1.09E-017** |
| 3 | 0.699 | **1.04E-017** | 0.374 | **1.73E-017** | 0.662 | **2.23E-017** |
| 4 | 0.244 | **1.04E-017** | 0.370 | **2.34E-017** | 0.735 | **2.36E-017** |
| 5 | 0.584 | **1.40E-017** | 0.748 | **2.34E-017** | 0.789 | **7.23E-017** |
| 6 | 0.688 | **1.17E-017** | 0.900 | **2.20E-017** | 0.792 | **5.39E-017** |
| 7 | 0.639 | **1.17E-017** | 0.823 | **2.34E-017** | 0.756 | **1.36E-016** |
| 8 | 0.859 | **1.49E-017** | 0.872 | **4.02E-017** | 0.983 | **3.24E-016** |
| 9 | 0.782 | **1.67E-017** | 0.916 | **3.16E-017** | 0.820 | **4.56E-016** |
| 10 | 0.773 | **1.49E-017** | 0.928 | **2.98E-017** | 0.621 | **7.66E-017** |
| 11 | 0.205 | **1.40E-017** | 0.566 | **2.81E-017** | 0.464 | **1.53E-016** |
| 12 | 0.462 | **1.67E-017** | 0.961 | **3.16E-017** | 0.383 | **1.82E-016** |
| 13 | 0.562 | **1.58E-017** | 0.861 | **4.81E-017** | 0.655 | **1.45E-016** |
| 14 | 0.873 | **1.99E-017** | 0.806 | **2.98E-017** | 0.613 | **5.11E-016** |
| 15 | 0.806 | **1.99E-017** | 0.933 | **2.81E-017** | 0.691 | **6.06E-016** |
| 16 | 0.831 | **1.99E-017** | 0.915 | **4.53E-017** | 0.650 | **1.86E-015** |
| 17 | 0.790 | **1.99E-017** | 0.983 | **3.16E-017** | 0.670 | **1.26E-015** |
| 18 | 0.672 | **1.77E-017** | 0.823 | **2.98E-017** | 0.789 | **1.96E-015** |

Table 3.9: *p*-values computed from Rank-Sum test for different energy utilisation with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.35155 | **3.47E-015** | 0.38615 | **8.36E-015** | 0.76823 | **1.04E-017** |
| 1 | 0.50916 | **5.13E-015** | 0.27048 | **8.83E-015** | 0.71706 | **1.13E-017** |
| 2 | 0.07070 | **6.33E-015** | 0.31002 | **7.66E-015** | 0.45642 | **1.06E-017** |
| 3 | 0.43419 | **4.46E-015** | 0.95978 | **1.01E-014** | 0.60331 | **1.06E-017** |
| 4 | 0.41713 | **7.60E-014** | 0.57104 | **9.99E-014** | 0.24658 | **5.96E-016** |
| 5 | 0.35821 | **4.11E-012** | 0.92008 | **3.17E-012** | 0.54371 | **1.81E-014** |
| 6 | 0.92719 | **1.08E-010** | 0.20339 | **1.84E-010** | 0.48528 | **2.47E-012** |
| 7 | 0.98509 | **5.30E-010** | 0.19726 | **4.93E-010** | 0.26229 | **6.98E-012** |
| 8 | 0.87626 | **7.22E-008** | 0.07608 | **4.33E-008** | 0.08079 | **3.11E-008** |
| 9 | 0.57075 | **1.78E-006** | 0.13633 | **7.88E-007** | 0.11971 | **2.33E-006** |
| 10 | **0.04101** | **7.19E-006** | 0.11420 | **3.24E-007** | **0.00322** | **3.05E-005** |
| 11 | **0.00722** | **0.00037** | 0.19129 | **0.00001** | **8.42E-005** | **0.00153** |
| 12 | **0.00722** | **0.00412** | 0.45671 | **0.00001** | **0.00067** | **0.00929** |
| 13 | **0.00850** | **0.01963** | 0.60015 | **7.43E-006** | **0.00064** | **0.03024** |
| 14 | **0.04493** | **0.03068** | 0.35514 | **0.00014** | **0.00312** | 0.12263 |
| 15 | 0.30474 | **0.01726** | 0.19129 | **0.00048** | **0.00293** | 0.16889 |
| 16 | 0.66233 | **0.00336** | **0.03054** | **0.00059** | **0.00167** | 0.35722 |
| 17 | 0.72980 | **0.00570** | **0.01212** | **0.00179** | **0.00200** | 0.56461 |
| 18 | 0.75980 | **0.00568** | **0.00385** | **0.00215** | **0.00136** | 0.76604 |

Table 3.10: *p*-values computed from Rank-Sum test for different energy utilisation with 10s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.523 | **0.998** | 0.524 | **0.996** | 0.501 | **0.005** |
| 1 | 0.619 | **1.000** | 0.634 | **0.998** | 0.492 | **0.001** |
| 2 | 0.555 | **0.996** | 0.552 | **0.997** | 0.486 | **0.003** |
| 3 | 0.523 | **0.995** | 0.552 | **0.997** | 0.526 | **0.008** |
| 4 | 0.569 | **0.995** | 0.552 | **0.995** | 0.480 | **0.008** |
| 5 | 0.532 | **0.993** | 0.519 | **0.995** | 0.484 | **0.016** |
| 6 | 0.524 | **0.994** | 0.508 | **0.995** | 0.484 | **0.014** |
| 7 | 0.528 | **0.994** | 0.513 | **0.995** | 0.482 | **0.020** |
| 8 | 0.511 | **0.993** | 0.510 | **0.991** | 0.499 | **0.026** |
| 9 | 0.516 | **0.992** | 0.506 | **0.993** | 0.486 | **0.028** |
| 10 | 0.517 | **0.993** | 0.494 | **0.993** | 0.471 | **0.016** |
| 11 | 0.575 | **0.993** | 0.534 | **0.993** | 0.457 | **0.021** |
| 12 | 0.543 | **0.992** | 0.497 | **0.993** | 0.448 | **0.022** |
| 13 | 0.534 | **0.992** | 0.510 | **0.990** | 0.473 | **0.020** |
| 14 | 0.510 | **0.991** | 0.486 | **0.993** | 0.470 | **0.029** |
| 15 | 0.515 | **0.991** | 0.495 | **0.993** | 0.476 | **0.030** |
| 16 | 0.513 | **0.991** | 0.494 | **0.993** | 0.473 | **0.036** |
| 17 | 0.516 | **0.991** | 0.499 | **0.993** | 0.475 | **0.036** |
| 18 | 0.525 | **0.991** | 0.513 | **0.993** | 0.484 | **0.039** |

Table 3.11: *A*-values computed from Vargha-Delaney Test for different energy utilisation with 0.1s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows large effect size and in italic shows medium effect size.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.565 | **0.998** | 0.555 | **0.995** | 0.481 | **0.005** |
| 1 | 0.453 | **0.999** | 0.429 | **0.999** | 0.476 | **0.000** |
| 2 | 0.373 | **0.997** | 0.434 | **1.000** | 0.549 | **0.000** |
| 3 | 0.445 | **1.000** | 0.496 | **0.998** | 0.534 | **0.000** |
| 4 | 0.442 | **0.981** | 0.537 | **0.979** | 0.576 | **0.028** |
| 5 | 0.433 | **0.950** | 0.493 | **0.952** | 0.540 | **0.053** |
| 6 | 0.507 | **0.915** | 0.583 | **0.910** | 0.546 | **0.091** |
| 7 | 0.498 | **0.903** | 0.585 | **0.900** | 0.574 | **0.100** |
| 8 | 0.488 | **0.850** | 0.617 | **0.852** | 0.614 | **0.177** |
| 9 | 0.459 | **0.807** | 0.597 | **0.818** | 0.602 | **0.224** |
| 10 | 0.355 | **0.791** | 0.604 | **0.829** | *0.690* | **0.257** |
| 11 | *0.307* | **0.736** | 0.588 | **0.834** | **0.750** | *0.315* |
| 12 | *0.307* | *0.690* | 0.550 | **0.799** | *0.718* | 0.348 |
| 13 | *0.311* | 0.655 | 0.536 | **0.788** | *0.719* | 0.373 |
| 14 | 0.353 | 0.643 | 0.563 | **0.749** | *0.694* | 0.410 |
| 15 | 0.425 | 0.658 | 0.588 | **0.725** | *0.692* | 0.420 |
| 16 | 0.468 | *0.692* | 0.643 | **0.723** | *0.704* | 0.446 |
| 17 | 0.474 | *0.681* | *0.666* | *0.703* | *0.701* | 0.466 |
| 18 | 0.478 | *0.680* | *0.688* | *0.699* | *0.708* | 0.482 |

Table 3.12: *A*-values computed from Vargha-Delaney Test for different energy utilisation with 10s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The value in bold shows large effect size and in italic shows medium effect size.

Table 3.13: The total number of MAC packets transmitted and received are higher in AOMDV compared to AODV, NST and TinyAODV even when no error is introduced.

| Protocol | Normal | 10 failures |
|---|---|---|
| AOMDV | 53290 | 57905 |
| NST | **37334** | 36682 |
| AODV | 38181 | **22302** |
| Tiny-AODV | 38227 | 28655 |

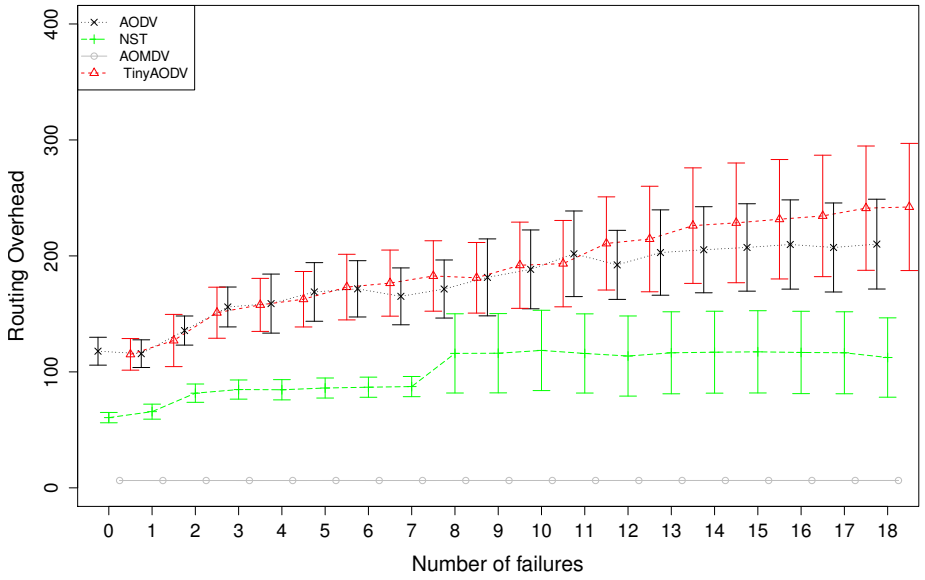
(5% to 7.6%), NST (4.8% to 6.8%) and Tiny-AODV (5% to 7.4%) increases with the number of failures. In contrast, the energy consumption for AOMDV decreases from 10% to 8% as more nodes are failed. The $A$-value tabulated in Table 3.12 shows that the energy consumed by NST is significantly less than AODV, TinyAODV and AOMDV when more than 16 nodes are failed.

Figure 3.7 shows the routing overhead generated by each of the routing protocols increases as the number of failures increases. In 3.7(a), the routing overhead generated by NST is half the number of routing overhead generated by AODV and TinyAODV with $p$-value $<<$ 0.0005 and $A$-value $<$ 0.27 (NST and AODV) and $< 0.29$ (NST and TinyAODV) in Table 3.16. NST has a lower routing packet due to its ability to retransmit during failure. Although the mean values shown in Figure 3.8(a) indicate the routing overhead for AODV is slightly lower than TinyAODV when 12 or more nodes are failed, the difference between AODV and TinyAODV indicated by the Rank-sum and Vargha-Delaney Test have shown that differences are not significant.

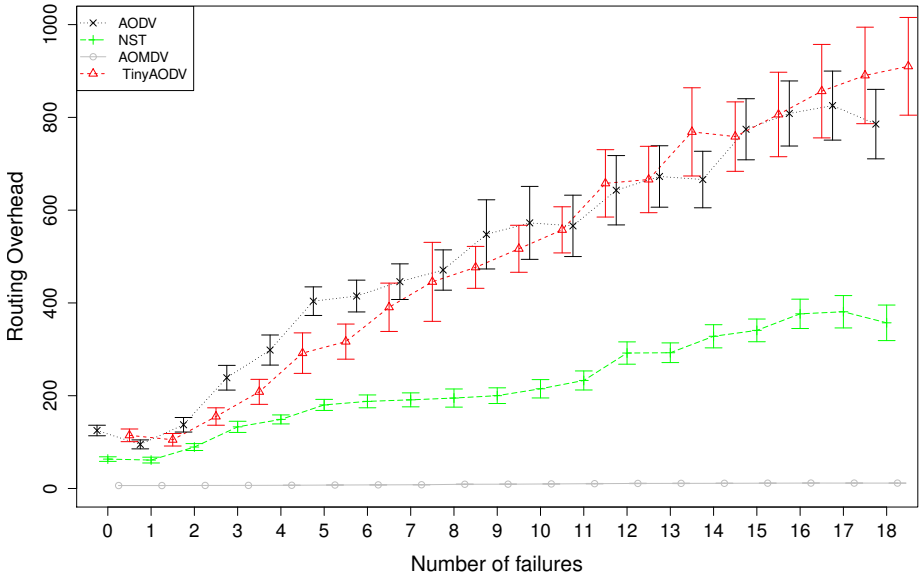
When the failure duration is increased to 10s, the number routing packet generated by AODV and TinyAODV increases faster than NST as a steeper gradient is observed in Figure 3.7(b). The routing overhead in NST is significantly lower than AODV and TinyAODV. This shows that the ability to retransmit the packet during failure in NST can reduce the routing overhead. The results in Figure 3.7(b) also show that the routing overhead generated by AODV and TinyAODV fluctuate between each other. When the number of failure is less than 2, the mean of the routing overhead is the same. Between 3 to 10 nodes, TinyAODV has a lower routing overhead than AODV. When more than 11 nodes have failed, the mean routing overhead is less than TinyAODV.

Further analysis of the simulation logs reveals that Local Discovery (LD) in AODV can generate more routing overhead than tinyAODV when no local node is available to forward the packets as shown in Figure 3.7(b) at 5 failure nodes. Additional 100 routing packets are sent and received by AODV when compared to TinyAODV (rank-sum test, $p$-value = 0.00338 and Vargha-Delaney test, $A$-value = 0.691). Hence, the results shows that the routing overhead generated by the routing protocols are affected by the failure condition occurring in the networks and the performance of the routing protocols are context dependent.

In terms of the packet delay, the average time required to deliver a packet is computed and shown in Figure 3.8. During 0.1s failure duration, the mean packet delay shown in Figure 3.8(a) for AODV, NST and TinyAODV slowly increases as the number of failures increases. The packet delay fluctuates between AODV, NST and TinyAODV as the number of failures increases. However, the differ-




(a) Routing Overhead for failure duration 0.1s




(b) Routing Overhead for failure duration 10s

Figure 3.7: The routing packets required to successfully delivered a packet are significantly higher in AODV and TinyAODV due to the lower PDR in AODV and TinyAODV. The routing overhead is higher for 10s failure durations as RDs are necessary to recover from the failing nodes.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **2.75E-008** | **3.29E-018** | **3.21E-008** | **4.83E-018** | 0.7225 | **4.83E-018** |
| 1 | **0.0001** | **7.15E-018** | **7.46E-007** | **1.06E-017** | 0.7291 | **7.16E-018** |
| 2 | **8.91E-007** | **4.84E-018** | **0.0001** | **1.06E-017** | 0.8314 | **7.16E-018** |
| 3 | **3.96E-007** | **4.84E-018** | **3.32E-005** | **1.06E-017** | 0.4639 | **7.16E-018** |
| 4 | **1.68E-005** | **4.84E-018** | **2.02E-005** | **1.06E-017** | 0.9482 | **7.16E-018** |
| 5 | **3.84E-006** | **4.84E-018** | **2.59E-005** | **1.06E-017** | 0.7618 | **7.16E-018** |
| 6 | **9.93E-007** | **4.84E-018** | **2.67E-005** | **1.06E-017** | 0.6313 | **7.16E-018** |
| 7 | **6.08E-006** | **4.83E-018** | **2.93E-005** | **1.06E-017** | 0.9425 | **7.15E-018** |
| 8 | **3.75E-005** | **4.84E-018** | **0.0003** | **1.06E-017** | 0.7183 | **7.16E-018** |
| 9 | **4.95E-005** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.8710 | **7.16E-018** |
| 10 | **2.71E-005** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.7427 | **7.16E-018** |
| 11 | **4.80E-006** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.6211 | **7.16E-018** |
| 12 | **4.43E-007** | **4.83E-018** | **0.0001** | **1.06E-017** | 0.3945 | **7.15E-018** |
| 13 | **1.37E-006** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.5207 | **7.16E-018** |
| 14 | **1.63E-006** | **4.83E-018** | **0.0005** | **1.06E-017** | 0.3709 | **7.15E-018** |
| 15 | **0.0001** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.5068 | **7.16E-018** |
| 16 | **1.50E-006** | **4.84E-018** | **0.0002** | **1.60E-017** | 0.4477 | **1.06E-017** |
| 17 | **1.82E-006** | **4.84E-018** | **0.0002** | **1.06E-017** | 0.5492 | **7.16E-018** |
| 18 | **7.16E-007** | **4.84E-018** | **0.0001** | **1.06E-017** | 0.4706 | **7.16E-018** |

Table 3.14: *p*-values computed from Rank-Sum test for different routing overhead with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **3.48E-007** | **2.62E-015** | **4.04E-006** | **4.44E-015** | 0.21122 | **4.83E-018** |
| 1 | **8.67E-005** | **4.44E-015** | **5.69E-005** | **7.64E-015** | 0.84331 | **1.06E-017** |
| 2 | **0.00065** | **4.46E-015** | **1.27E-005** | **7.66E-015** | 0.38601 | **1.06E-017** |
| 3 | **2.07E-005** | **4.46E-015** | **0.00037** | **7.66E-015** | 0.25435 | **1.06E-017** |
| 4 | **2.71E-006** | **7.64E-015** | **2.53E-005** | **7.65E-015** | 0.39118 | **1.06E-017** |
| 5 | **2.44E-011** | **1.33E-014** | **3.46E-005** | **1.33E-014** | **0.00338** | **1.06E-017** |
| 6 | **1.18E-009** | **7.66E-015** | **3.07E-008** | **7.66E-015** | 0.12199 | **1.06E-017** |
| 7 | **1.43E-011** | **1.33E-014** | **7.49E-009** | **7.65E-015** | 0.19350 | **1.06E-017** |
| 8 | **1.24E-010** | **1.33E-014** | **5.40E-012** | **7.66E-015** | 0.83322 | **1.06E-017** |
| 9 | **2.29E-010** | **7.66E-015** | **3.60E-013** | **7.66E-015** | 0.72558 | **1.06E-017** |
| 10 | **1.86E-011** | **1.33E-014** | **6.37E-014** | **7.66E-015** | 0.69766 | **1.06E-017** |
| 11 | **2.59E-011** | **4.20E-014** | **7.66E-013** | **7.66E-015** | 0.40165 | **1.06E-017** |
| 12 | **2.56E-009** | **4.20E-014** | **7.67E-009** | **7.66E-015** | 0.82592 | **1.06E-017** |
| 13 | **2.25E-011** | **4.20E-014** | **1.68E-010** | **7.66E-015** | 0.46211 | **1.06E-017** |
| 14 | **7.86E-010** | **4.20E-014** | **6.58E-009** | **2.35E-014** | 0.28843 | **1.06E-017** |
| 15 | **7.70E-011** | **4.20E-014** | **0.00001** | **7.66E-015** | 0.86991 | **1.06E-017** |
| 16 | **3.75E-009** | **2.35E-014** | **2.31E-007** | **1.33E-014** | 0.95809 | **1.06E-017** |
| 17 | **1.61E-008** | **2.35E-014** | **1.02E-007** | **1.33E-014** | 0.87646 | **1.60E-017** |
| 18 | **8.94E-008** | **1.33E-014** | **2.66E-009** | **1.33E-014** | 0.47019 | **1.06E-017** |

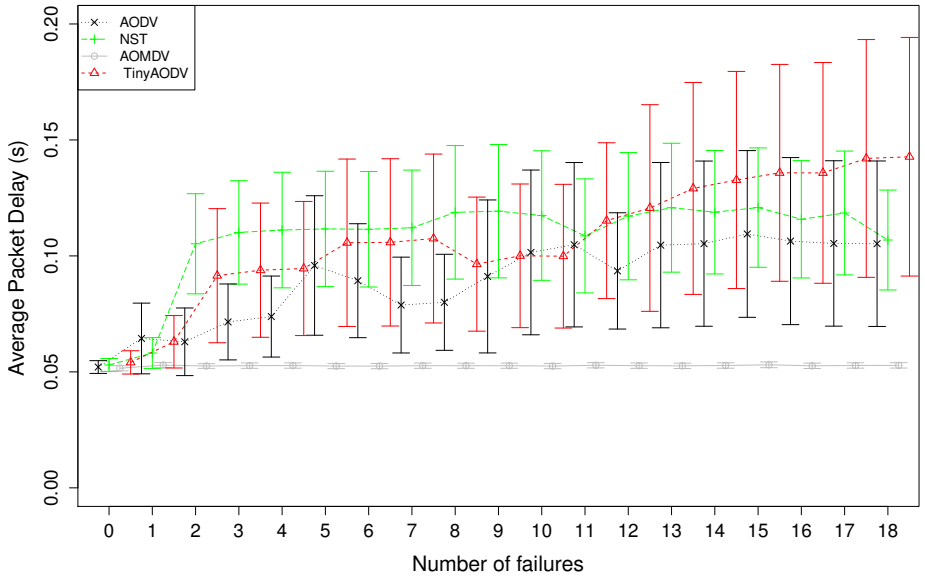
Table 3.15: *p*-values computed from Rank-Sum test for different routing overhead with 10s failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **0.179** | **1.000** | **0.181** | **1.000** | 0.521 | **0.000** |
| 1 | **0.196** | **1.000** | **0.210** | **1.000** | 0.521 | **0.000** |
| 2 | **0.212** | **1.000** | **0.253** | **1.000** | 0.513 | **0.000** |
| 3 | **0.203** | **1.000** | **0.258** | **1.000** | 0.543 | **0.000** |
| 4 | **0.248** | **1.000** | **0.251** | **1.000** | 0.496 | **0.000** |
| 5 | **0.229** | **1.000** | **0.254** | **1.000** | 0.518 | **0.000** |
| 6 | **0.213** | **1.000** | **0.255** | **1.000** | 0.528 | **0.000** |
| 7 | **0.235** | **1.000** | **0.256** | **1.000** | 0.504 | **0.000** |
| 8 | **0.258** | **1.000** | *0.291* | **1.000** | 0.521 | **0.000** |
| 9 | **0.262** | **1.000** | *0.286* | **1.000** | 0.510 | **0.000** |
| 10 | **0.254** | **1.000** | *0.284* | **1.000** | 0.520 | **0.000** |
| 11 | **0.232** | **1.000** | *0.280* | **1.000** | 0.529 | **0.000** |
| 12 | **0.204** | **1.000** | **0.269** | **1.000** | 0.550 | **0.000** |
| 13 | **0.217** | **1.000** | *0.280* | **1.000** | 0.538 | **0.000** |
| 14 | **0.219** | **1.000** | *0.298* | **1.000** | 0.553 | **0.000** |
| 15 | **0.217** | **1.000** | *0.280* | **1.000** | 0.539 | **0.000** |
| 16 | **0.216** | **1.000** | *0.281* | **1.000** | 0.545 | **0.000** |
| 17 | **0.220** | **1.000** | *0.280* | **1.000** | 0.536 | **0.000** |
| 18 | **0.209** | **1.000** | *0.275* | **1.000** | 0.543 | **0.000** |

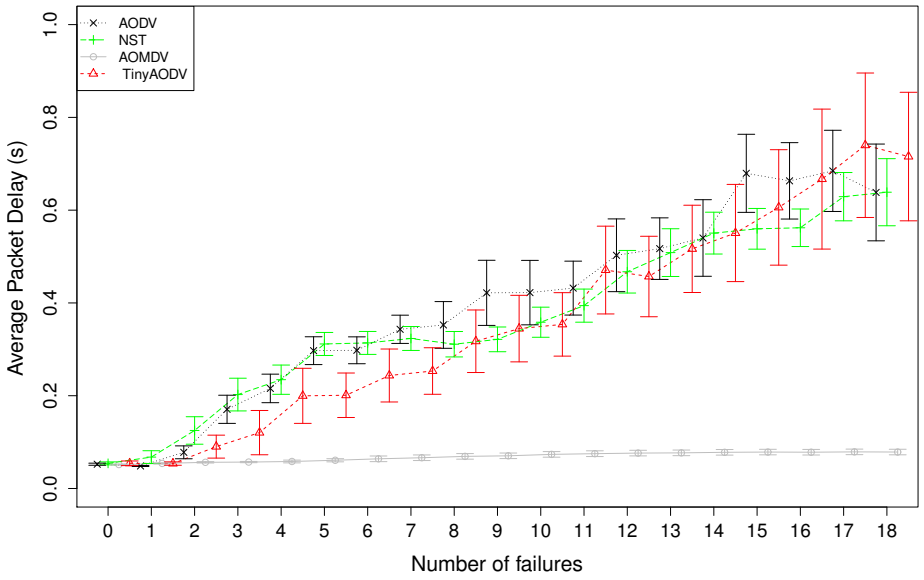
Table 3.16: *A*-values computed from Vargha-Delaney Test for different routing overhead with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows large effect size and in italic shows medium effect size.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | **0.148** | **1.000** | **0.207** | **1.000** | 0.580 | **0.000** |
| 1 | **0.225** | **1.000** | **0.240** | **1.000** | 0.487 | **0.000** |
| 2 | **0.266** | **1.000** | **0.227** | **1.000** | 0.443 | **0.000** |
| 3 | **0.212** | **1.000** | **0.274** | **1.000** | 0.575 | **0.000** |
| 4 | **0.169** | **1.000** | **0.225** | **1.000** | 0.556 | **0.000** |
| 5 | **0.072** | **1.000** | **0.235** | **1.000** | *0.691* | **0.000** |
| 6 | **0.106** | **1.000** | **0.159** | **1.000** | 0.601 | **0.000** |
| 7 | **0.070** | **1.000** | **0.144** | **1.000** | 0.585 | **0.000** |
| 8 | **0.086** | **1.000** | **0.090** | **1.000** | 0.514 | **0.000** |
| 9 | **0.093** | **1.000** | **0.076** | **1.000** | 0.523 | **0.000** |
| 10 | **0.072** | **1.000** | **0.064** | **1.000** | 0.474 | **0.000** |
| 11 | **0.069** | **1.000** | **0.072** | **1.000** | 0.445 | **0.000** |
| 12 | **0.104** | **1.000** | **0.138** | **1.000** | 0.485 | **0.000** |
| 13 | **0.068** | **1.000** | **0.108** | **1.000** | 0.452 | **0.000** |
| 14 | **0.090** | **1.000** | **0.137** | **1.000** | 0.429 | **0.000** |
| 15 | **0.077** | **1.000** | **0.150** | **1.000** | 0.511 | **0.000** |
| 16 | **0.108** | **1.000** | **0.174** | **1.000** | 0.504 | **0.000** |
| 17 | **0.121** | **1.000** | **0.164** | **1.000** | 0.489 | **0.000** |
| 18 | **0.140** | **1.000** | **0.135** | **1.000** | 0.452 | **0.000** |

Table 3.17: *A*-values computed from Vargha-Delaney Test for different routing overhead with 10s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows large effect size and in italic shows medium effect size.




(a) Average Packet Delay for failure duration 0.1s




(b) Average Packet Delay for failure duration 10s

Figure 3.8: The packet delay in TinyAODV is lower than AODV and NST-AODV as packet are dropped when a failure is detected in the network. It is more appropriate to perform global route discovery when there is no route available locally to forward to packet in order to allow the next packet to be delivered quickly.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.6911 | **4.70E-018** | 0.6761 | **9.31E-018** | 0.9890 | **9.87E-018** |
| 1 | **0.0437** | **7.17E-018** | **0.0224** | **1.44E-017** | 0.8910 | **8.59E-018** |
| 2 | 0.3500 | **8.27E-018** | 0.3704 | **1.73E-017** | 0.8202 | **1.09E-017** |
| 3 | 0.6985 | **1.04E-017** | 0.3741 | **1.73E-017** | 0.6624 | **2.23E-017** |
| 4 | 0.2438 | **1.04E-017** | 0.3704 | **2.34E-017** | 0.7345 | **2.36E-017** |
| 5 | 0.5842 | **1.40E-017** | 0.7475 | **2.34E-017** | 0.7895 | **7.23E-017** |
| 6 | 0.6880 | **1.17E-017** | 0.8998 | **2.20E-017** | 0.7923 | **5.39E-017** |
| 7 | 0.6390 | **1.17E-017** | 0.8228 | **2.34E-017** | 0.7564 | **1.36E-016** |
| 8 | 0.8590 | **1.49E-017** | 0.8721 | **4.02E-017** | 0.9827 | **3.24E-016** |
| 9 | 0.7817 | **1.67E-017** | 0.9164 | **3.16E-017** | 0.8202 | **4.56E-016** |
| 10 | 0.7735 | **1.49E-017** | 0.9275 | **2.98E-017** | 0.6211 | **7.66E-017** |
| 11 | 0.2046 | **1.40E-017** | 0.5660 | **2.81E-017** | 0.4639 | **1.53E-016** |
| 12 | 0.4620 | **1.67E-017** | 0.9609 | **3.16E-017** | 0.3826 | **1.82E-016** |
| 13 | 0.5625 | **1.58E-017** | 0.8611 | **4.81E-017** | 0.6546 | **1.45E-016** |
| 14 | 0.8730 | **1.99E-017** | 0.8065 | **2.98E-017** | 0.6135 | **5.11E-016** |
| 15 | 0.8063 | **1.99E-017** | 0.9331 | **2.81E-017** | 0.6915 | **6.06E-016** |
| 16 | 0.8314 | **1.99E-017** | 0.9151 | **4.53E-017** | 0.6498 | **1.86E-015** |
| 17 | 0.7899 | **1.99E-017** | 0.9832 | **3.16E-017** | 0.6703 | **1.26E-015** |
| 18 | 0.6724 | **1.77E-017** | 0.8228 | **2.98E-017** | 0.7895 | **1.96E-015** |

Table 3.18: *p*-values computed from Rank-Sum test for different average packet delay with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | *N:A* | *N:M* | *N:T* | *A:M* | *A:T* | *M:T* |
| 0 | 0.74743 | 0.69496 | 0.50028 | 0.27959 | 0.96439 | 0.12393 |
| 1 | 0.29086 | 0.05995 | 0.55199 | **0.00083** | 0.64127 | **0.00303** |
| 2 | 0.09421 | 0.40863 | 0.06426 | 0.11962 | 0.94404 | 0.07709 |
| 3 | 0.24361 | **0.00010** | **0.00444** | **0.03621** | 0.11971 | 0.14628 |
| 4 | 0.29638 | **1.01E-007** | **0.02360** | 0.00015 | 0.20339 | **0.00081** |
| 5 | 0.70230 | **2.05E-013** | **4.36E-005** | **3.54E-011** | **0.00063** | **3.55E-005** |
| 6 | 0.86934 | **1.71E-013** | **0.00106** | **2.53E-012** | **0.00756** | **0.00001** |
| 7 | 0.60498 | **4.33E-013** | **0.00932** | **1.83E-013** | **0.00366** | **4.71E-007** |
| 8 | 0.59630 | **6.56E-011** | 0.08625 | **1.60E-013** | **0.03327** | **2.35E-008** |
| 9 | 0.42419 | **1.61E-012** | 0.11527 | **1.22E-011** | 0.07915 | **1.32E-007** |
| 10 | 0.76018 | **1.03E-012** | 0.16283 | **8.43E-012** | 0.15466 | **2.33E-007** |
| 11 | 0.96359 | **3.14E-012** | 0.43875 | **4.36E-013** | 0.37075 | **5.22E-008** |
| 12 | 0.73939 | **2.23E-012** | 0.25194 | **1.16E-012** | 0.24275 | **1.37E-007** |
| 13 | 0.72956 | **2.08E-012** | 0.35514 | **9.60E-013** | 0.42853 | **9.36E-008** |
| 14 | 0.34879 | **2.95E-013** | 0.29135 | **3.49E-012** | 0.52859 | **2.17E-008** |
| 15 | 0.35690 | **3.40E-013** | 0.36568 | **2.39E-013** | 0.15193 | **5.07E-007** |
| 16 | 0.42224 | **2.13E-013** | 0.46658 | **1.64E-012** | 0.23668 | **5.64E-008** |
| 17 | 0.80949 | **2.13E-013** | 0.58146 | **2.04E-014** | 0.39188 | **5.76E-008** |
| 18 | 0.45952 | **3.32E-010** | 0.84469 | **1.00E-010** | 0.81488 | **1.17E-008** |

Table 3.19: *p*-values computed from Rank-Sum test for different average packet delay with 10s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows statistical significance.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | ***N:A*** | ***N:M*** | ***N:T*** | ***A:M*** | ***A:T*** | ***M:T*** |
| 0 | 0.535 | 0.461 | 0.539 | 0.399 | 0.496 | 0.595 |
| 1 | 0.513 | 0.402 | 0.508 | 0.339 | 0.490 | 0.615 |
| 2 | 0.616 | 0.565 | 0.581 | 0.385 | 0.463 | 0.527 |
| 3 | 0.608 | 0.589 | 0.572 | 0.434 | 0.469 | 0.494 |
| 4 | 0.589 | 0.584 | 0.570 | 0.473 | 0.482 | 0.506 |
| 5 | 0.594 | 0.629 | 0.584 | 0.484 | 0.495 | 0.505 |
| 6 | 0.590 | 0.615 | 0.571 | 0.475 | 0.480 | 0.480 |
| 7 | 0.614 | 0.614 | 0.574 | 0.446 | 0.462 | 0.485 |
| 8 | 0.599 | 0.625 | 0.579 | 0.482 | 0.484 | 0.490 |
| 9 | 0.611 | 0.637 | 0.588 | 0.474 | 0.475 | 0.492 |
| 10 | 0.588 | 0.619 | 0.583 | 0.487 | 0.493 | 0.503 |
| 11 | 0.575 | 0.609 | 0.557 | 0.498 | 0.489 | 0.483 |
| 12 | 0.572 | 0.616 | 0.574 | 0.503 | 0.509 | 0.497 |
| 13 | 0.599 | 0.641 | 0.579 | 0.488 | 0.486 | 0.480 |
| 14 | 0.599 | 0.636 | 0.583 | 0.477 | 0.496 | 0.498 |
| 15 | 0.602 | 0.634 | 0.572 | 0.464 | 0.479 | 0.482 |
| 16 | 0.594 | 0.625 | 0.569 | 0.476 | 0.484 | 0.484 |
| 17 | 0.584 | 0.618 | 0.564 | 0.477 | 0.484 | 0.486 |
| 18 | 0.594 | 0.629 | 0.571 | 0.464 | 0.478 | 0.485 |

Table 3.20: *A*-values computed from Vargha-Delaney Test for different average delay with 0.1s failure duration where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows large effect size and in italic shows medium effect size.

| *Number* | **Routing Protocols** | | | | | |
|---|---|---|---|---|---|---|
| *Failures* | ***N:A*** | ***N:M*** | ***N:T*** | ***A:M*** | ***A:T*** | ***M:T*** |
| 0 | 0.523 | 0.475 | 0.543 | 0.431 | 0.503 | 0.589 |
| 1 | 0.574 | 0.380 | 0.539 | 0.285 | 0.469 | 0.673 |
| 2 | 0.618 | 0.553 | 0.620 | 0.400 | 0.505 | 0.603 |
| 3 | 0.582 | **0.748** | *0.682* | 0.635 | 0.602 | 0.415 |
| 4 | 0.574 | **0.843** | 0.647 | **0.745** | 0.583 | *0.304* |
| 5 | 0.528 | **0.977** | **0.761** | **0.930** | *0.721* | **0.259** |
| 6 | 0.512 | **0.974** | *0.710* | **0.950** | *0.673* | **0.215** |
| 7 | 0.463 | **0.970** | *0.670* | **0.973** | *0.688* | **0.206** |
| 8 | 0.462 | **0.924** | 0.613 | **0.975** | 0.638 | **0.174** |
| 9 | 0.443 | **0.954** | 0.603 | **0.936** | 0.615 | **0.192** |
| 10 | 0.478 | **0.963** | 0.592 | **0.939** | 0.593 | **0.198** |
| 11 | 0.496 | **0.961** | 0.552 | **0.966** | 0.559 | **0.182** |
| 12 | 0.525 | **0.965** | 0.577 | **0.957** | 0.577 | **0.192** |
| 13 | 0.526 | **0.965** | 0.563 | **0.959** | 0.552 | **0.188** |
| 14 | 0.570 | **0.983** | 0.571 | **0.956** | 0.542 | **0.173** |
| 15 | 0.433 | **0.982** | 0.561 | **0.971** | 0.594 | **0.207** |
| 16 | 0.441 | **0.981** | 0.549 | **0.958** | 0.578 | **0.183** |
| 17 | 0.518 | **0.981** | 0.537 | **0.996** | 0.557 | **0.181** |
| 18 | 0.554 | **0.908** | 0.513 | **0.920** | 0.516 | **0.167** |

Table 3.21: *A*-values computed from Vargha-Delaney Test for different average delay with 10 failure durations where N=NST, A=AODV, M=AOMDV and T=TinyAODV. The values in bold shows large effect size and in italic shows medium effect size.

ences in packet delay between AODV, NST and TinyAODV are not significant as indicated by the $p$- and $A$-value in Table 3.20 and 3.21.

### 3.2.2 Discussion

The performance results observed are clearly context dependent. The simulation results suggest that different failure characteristics in terms of duration and size can affect the routing behaviour, network reliability and efficiency. Response taken by the routing protocol to overcome failure and its parameter can also affect the network performance. To improve the performance of the networks, routing approach taken needs to adapt to the operating condition.

## 3.3 Summary

The AODV have been enhanced over the last few years to improve its performance. The four common routing protocols applied in WSNs were evaluated. From our studies, the performances of AODV have degraded very quickly as we increased the number of failures. More packets are dropped in AODV than NST due to network congestion created by RD packets. The energy consumption and delay are higher in AODV and NST-AODV than TinyAODV at longer failure duration. Although optimistic retransmission in NST can improve the probability of packet transmission during transient failure, it utilises the resources unnecessary when the failing duration of the next hop neighbour is longer than the retransmission period assigned. When we increased the failure duration and the number of failures, the packet latency and energy utilisation in NST begins to increase as each packet is queued while the node attempts to retransmit the failed packet. The overall performance in NST starts to degrade as it reverts to less optimistic local repair in AODV. These results have provided a motivation for a mixed-hybrid routing approach to be investigated in the next chapter.

CHAPTER

# FOUR

# Achieving Dependability using Multi-modal Network Protocol

In this chapter, we propose a multi-modal approach switching between different protocols to improve the robustness and dependability of the WSNs. The motivation for the work is introduced in Section 4.1. To investigate the robustness and effectiveness of the multi-modal concept to improve dependability, a Multi-modal Routing Protocol (MRP) is proposed and presented in Section 4.2. To evaluate the robustness and scalability of MRP, we have deployed two types of networks (indoor and outdoor) in simulation to measure the performance of MRP against AODV and NST-AODV. Using the data collected from the indoor simulations, we investigate the robustness of the MRP by varying the failure durations and the number of failures. We present the experimental setup and analyse the results in Section 4.3 and 4.4. In Section 4.5, we demonstrate further experiments to evaluate the scalability of MRP by increasing the number of nodes in the networks based on an outdoor application. Using the data collected, the performances are reported for different network sizes. We summarise the results observed from the experiments in Section 4.6.

## 4.1 Motivation

In the real world deployment of WSNs, individual nodes are usually arranged in a pre-defined static topology to ensure the radio connectivity between nodes and avoiding holes (Wagner, 2010; Togami et al., 2012). Due to the over-utilisation

of the unlicensed radio channel by other wireless devices, the radio communication between the nodes can be inflected by radio frequency interference generated by these devices that can occur anywhere, anytime. Zhou et al. (2004) have demonstrated that these radio irregularity can degrade the reliability of the WSNs and disrupt the routing protocol. Huo et al. (2009) show that these interferences may cause significant packet losses of between 10% to 50%. Although the IEEE 802.15.4 MAC standard (IEEE, 2006b) attempts to address this problem by using frame acknowledgement to retransmit the packet immediately after failure, the retransmission is not always successful due to the temporal characteristic of the high radio signal noise generated by the interfering device. Srinivasan et al. (2010) report that a high radio emitting signal from 802.11 device can create spatial and temporal correlated packet losses as the noise may also affect its neighbouring nodes. Hence, a group of local nodes may fail depending on the characteristics of the noise generated (Lee et al., 2007b). Environmental effects such as bad weather and blocking objects is unavoidable and can affect the wireless channel (Szewczyk et al., 2004). Therefore, it is desirable to have a robust network protocol that can tolerate these negative effects.

We propose a multimodal approach that toggles between different operating states using a set of rules according to the changing environment. We formalise the following hypothesis to answer the first research question in this thesis:

> *The reliability of WSNs in term of Packet Delivery rate can be improved by integrating and switching between routing protocols to function according to its operating environments.*

## 4.2 Multi-modal Protocol

In this section, we introduce the concept of multimodal to support different routing routing in WSNs.

### 4.2.1 Multimodal Routing Protocol

In order to tolerate failures with different densities and durations, the operating protocol should be able to adapt and function according to the state of the networks. As highlighted in Chapter 3, operating in one routing protocol may not be able to overcome different failures. It is necessary combine a number of routing protocols and activate one of the protocol according to the temporal condition it will work best. The way to achieve this is to apply a multi-modal concept to allow multiple routing protocols to be implemented in the node. Different nodes

are allowed to run different routing protocols in parallel in order to achieve the dependability level required by the application.

To show that the multimodal approach is more reliable and robust than single mode approach in dynamic networks, we define the four main objectives in order to address the research question defined in Chapter 1.1.4 as:

- Objective E4-1: to demonstrate that the multimodal approach can achieve the desired network reliability (90%) compared to a single mode.
- Objective E4-2: to demonstrate that the multimodal approach does not consume more energy resources than a single mode.
- Objective E4-3: to demonstrate that the multimodal approach has a lower communication overhead than a single mode.
- Objective E4-4: to demonstrate that the multimodal approach has a lower packet delay than a single mode.

### 4.2.2 Design of Multi-modal Routing Protocol

In this section, we investigate the multimodal approach applied on the network protocol and propose the MRP based a spectrum of routing protocols namely: AODV, NST, and TinyAODV. The current system is based on a spectrum of reactive routing protocols. NST deals well with short-term node failures as it attempts a number of re-transmissions before performing local re-routing. In contrast, AODV handles long-term failures well as it gives us local re-routing before trying global re-routing. TinyAODV takes a pessimistic recovery approach to overcome long-term spatial failure by performing a global re-routing without local repair. If we can predict whether we have a short or long-term node failure, then we can choose the best (in terms of PDR, energy usage and latency) protocol in the current context.

By incorporating routing protocols with the reactive property, the switching delay and service disruption reported by Figueredo et al. (2005) can be overcome. A switching module is used to switch between the AODV, the NST and the TinyAODV to improve the network reliability and efficiency. Each node can operate autonomously in either one of the routing modes depending on the probability (success rate) of the routing protocol to overcome a failure.

The MRP consists of Route Selection Module (RSM), a set of routing protocols, conditional table and thresholds as illustrates in Figure 4.1. The RSM is implemented in the network layer for the RSM to access and activate the routing

protocol immediately. To detect failure, MRP relies on the Link Layer to report for any transmission failure. In the AODV protocol proposed by Perkins et al. (2003), the routing protocol can either depend on the *Hello* packet or the LLN for detect network failure. In MRP, the *Hello* packet is not used as the results in Chapter 3 have shown that enabling hello packet can severely reduce the network performance. Instead, MRP depends on the LLN to detect and alert any packet missing an acknowledgement. A single-hop packet acknowledgement is proposed as a higher data delivery probability (above 90%) can be achieved (Zhao et al., 2013).

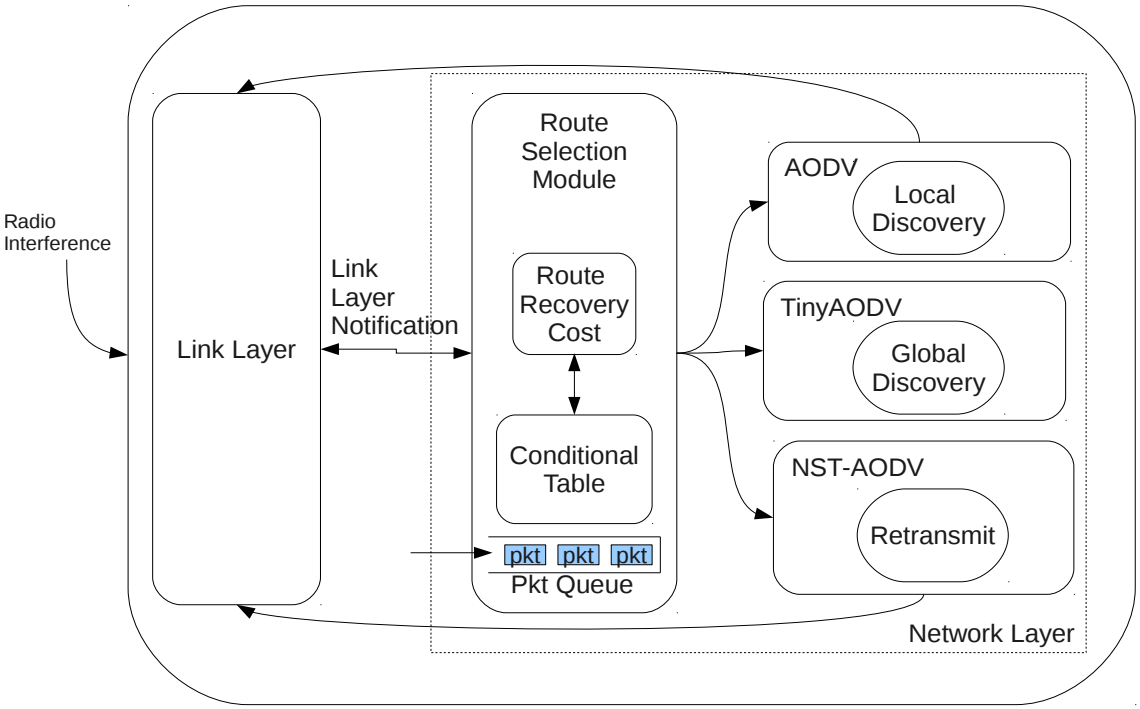



Figure 4.1: The Architecture of Multimodal Routing Protocol

To allow intermediate nodes to make localised decisions to switch between AODV, NST or TinyAODV, a self-switching route mechanism is proposed. The switching decision is made autonomously and independently in the RSM based on the success rate of the previous attempts of the routing protocol to overcome failures. Each routing protocol can use different approaches to overcome failures: NST uses the optimistic approach to ReTransmission (RT), AODV uses Local Discovery (LD) and TinyAODV uses the pessimistic approach to initiate Global Discovery (GD). Once a routing decision is activated, the RSM module will wait for the feedback from the link layer, and evaluate the effectiveness of that decision. Based on the evaluation, it updates the cost to execute to routing protocol in the response table.

In order to manage the switching mechanism, each route recovery approach

is assigned with a cost to assist in the switching decision namely:

- $Cost_{RT} \in \mathbb{Z}$ corresponds to the cost of RT in NST,
- $Cost_{LD} \in \mathbb{Z}$ corresponds to the cost of LD in AODV,
- $Cost_{GD} \in \mathbb{Z}$ corresponds to the cost of GD in TinyAODV,

Using the rules specified in the conditional table, a routing protocol is only selected when

$$\textbf{Routing} = \begin{cases} NST, & \text{if } Cost_{RT} \leq RT_{max} \text{ or } Cost_{RT} \leq Cost_{LD}, \\ AODV, & \text{if } Cost_{LD} \leq LD_{max} \text{ and } Cost_{LD} \leq Cost_{GD}, \\ TinyAODV, & \text{if } Cost_{GD} \leq GD_{max}. \end{cases} \tag{4.1}$$

where $RT_{max}$ is the maximum number of attempts for RT,
$LD_{max}$ is the maximum number of attempts for LD, and
$GD_{max}$ is the maximum number of attempts for GD.

During startup, the variables ($Cost_{RT}$, $Cost_{LD}$, and $Cost_{GD}$) that control the switching between the routing protocols are initialised to zero. Each routing protocol is assigned with a threshold value ($RT_{max}$, $LD_{max}$, $GD_{max}$). When the application wants to send a data packet, the RSM look-ups the routing table for the available route to forward the packet. If a route is not available, GD is initiated. If a route is available, the node will transmit the packet. If a packet cannot be sent successfully, a LLN is issued by the link layer. Based on the LLN receive, the RSM updates and adjusts the routing cost according to the protocol executed. For each failure, the cost of the routing function executed is increased by one until it reaches its threshold value ($RT_{max}$, $LD_{max}$, $GD_{max}$). Alternatively, the routing costs are reduced by one for each successful transmission until it is zero. Based on Equation 4.1, a routing protocol is only executed if its cost is below its threshold. A state diagram is shown in Figure 4.2, highlighting different recovery states based on AODV and NST-AODV routing protocols.

During each unsuccessful packet transmission, the RSM evaluates and selects the routing protocol with the lowest cost and is below the threshold value ($RT_{max}$, $LD_{max}$, $GD_{max}$). It is a common practise in network community to set the number of maximum attempt to 3. The IEEE 802.15.4 standard recommends that the maximum number of retries to be set to 3 as too many retransmissions may aggravate

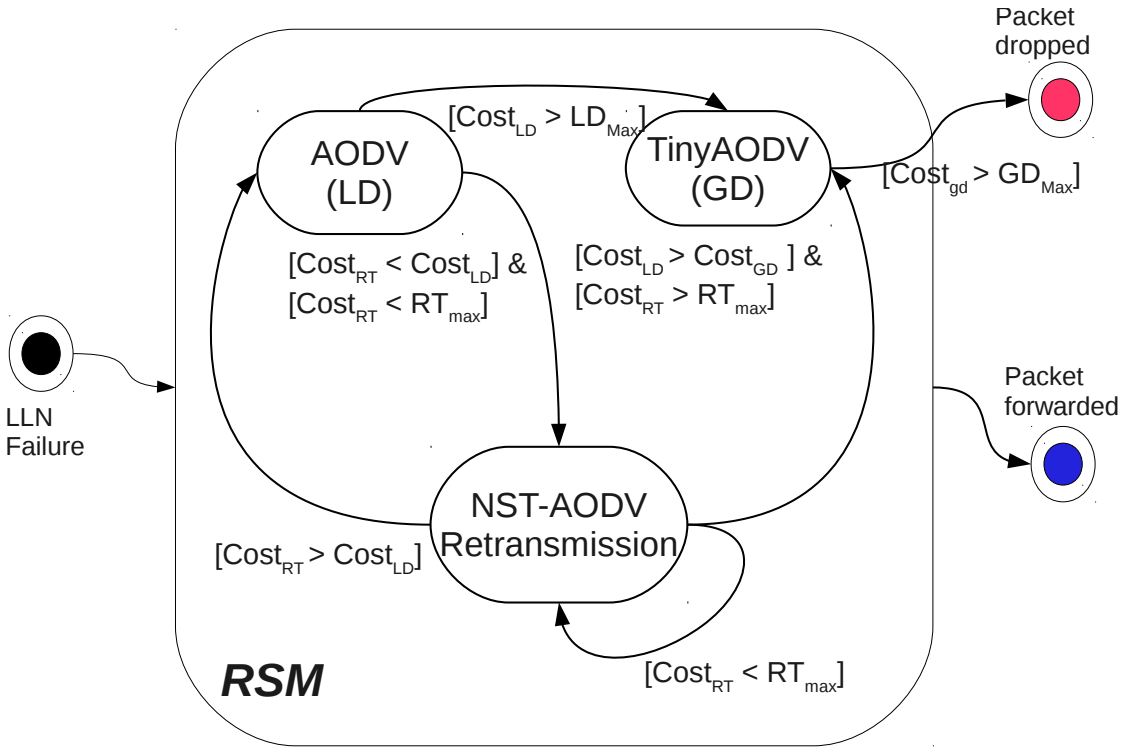



Figure 4.2: Flow Diagram for MRP during failures.

the network (IEEE, 2006a). Zhao et al. (2013) have observed an increase in packet delivery probability when the number of packets retransmission were between 3 to 6. Hence, the same maximum thresholds ($RT_{max}$ ,$LD_{max}$, $GD_{max}$ = 3) are set in all the routing protocols using the recommended retransmission. Although the parameters selected may not be the best value, it should be sufficient for us to investigate and compare the performance of the multimodal against single-modal approach. As these parameters can be highly dependent on the network utilisation and condition, we acknowledge that there may a need to adapt and apply reinforced learning to tune these thresholds dynamically to work according to the environment. Hence, it has been proposed as a future research work.

Once a routing mode is selected, the forwarding node is set to wait for a random number of backoff periods $Tx_{backoff} \in [0, B]$ where $B = (2^n - 1)$ and $n(numberofattempts) \in [0, 4]$ specified by the IEEE (2006a) for IEEE 802.15.4 MAC in order to overcome packet collision due to network congestion. Once the backoff period expires, the selected routing protocol is activated. The activation of the routing protocol is repeated until it reaches its threshold level. Once this threshold level is reached, the failed packet will be stored in the node's transmission buffer and the MRP switches the next lowest routing protocol. When a new packet is received during that time, the node will attempt to forward the packet

using its current routing mode. The MRP algorithm is outlined in Algorithm 1.

**Algorithm 1:** Multimodal Routing Protocol Algorithm

**Input** : $P_s$:Packet Send
**Input** : $P_{ack}$:Packet Acknowledge
**Output**: Response Action

```
1  while Packet Buffer is not Empty do
2      Send Packet P_s and wait for acknowledgement P_ack
3      if P_ack is received then
4          Decrease RT_cost
5      else if RT_cost < RT_max or RT_cost < LD_cost and Route is valid then
6          Retransmit                                          } Switch To
7          Increase RT_cost                                    } NST.
8      else if LD_cost < LD_max and LD_cost < GD_cost then
9          Perform Local Discovery                             } Switch
10         Increase LD_cost                                    } To
11         if Route Discovery is Successful then               } AODV(LD).
12             Decrease LD_cost, RT_cost                       }
13     else if GD_cost < GD_max then
14         Invalidate Route
15         Send Error for Global                               } Switch
           Discovery                                           } To
16         Increase GD_cost                                    } AODV(GD).
17     if Timeout then
18         Reinitialised
```

### 4.2.3 Discussion

In this section, we have presented a multi-mode routing approach to provide different routing strategies depending on current network conditions. Three reactive routing protocols (NST, AODV and TinyAODV) have been integrated to minimise disruption of network services while trying to increase the reliability of packet delivery and lower energy consumption. The reactive protocols have been proposed to reduce the switching overhead and a linear threshold is applied to control the switching between the routing protocols. Although the current MRP only supports three reactive routing protocols, additional reactive routing protocol can be added to the MRP to tolerate new failure pattern. For instance, an adaptive power transmission routing proposed by Lin et al. (2006) can be applied to overcome failure that has short duration and strong intensity by increasing the transmission range of the node above the interference signal. Other reactive routing such as DSR and AOMDV can be used to handle failure that has long duration and weak intensity.

In order to demonstrate the ability of the MRP to tolerate with different operating environments, the robustness and scalability of MRP based on both indoor and outdoor scenarios are evaluated using three sets of simulated experiments in the following sections.

## 4.3 On Robustness: 1. The effects of failure duration

To compare the robustness of the MRP, NST and AODV routing protocols, we analyse the performance of MRP against AODV and NST under the influence of different failure durations using the data obtained from simulation. In order to investigate the robustness of the routing protocols, we vary the failure durations by failing a node along the active path with an increasing failure population to simulate the effect on different types of interference sources affecting the number of nodes along the route. A number of random nodes in WSNs can fail for a period of time depending on the duration and strength of the interference source. We present the application scenario and the simulation setup and analyse the results using statistical tools in Section 4.3.5.

### 4.3.1 Application Scenario

The application scenario used in this experiment is based on an indoor patient monitoring system applied in hospital (Chipara et al., 2010). In order to collect vital data, sensor nodes are attached on the patient which periodically capture medical data such as heart rate and blood pressure. For example, the oxygen level in the patient blood must be collected and delivered to the end user every 5 to 10 seconds for a pulse oximetry application (Chipara et al., 2010). In this application, we assume a periodic data stream which is send across multiple nodes using a multi-hop routing protocol as the control station can be far away from the patients room as shown in Figure 4.3. We believe this periodic transmission is applicable across most applications as the health of the WSNs need to be checked using periodic heartbeat packets send across the network even in applications that are event driven (Xiao et al., 2010).

### 4.3.2 Network Setup

The nodes in WSNs are usually deployed in a flat-based topology where each node are usually placed within the transmission range of the neighbouring nodes. In this experiment, 51 static nodes are deployed across the simulated environment and positioned at the top of the wall as shown in Figure 4.3. The nodes are

placed 10m apart with the transmission range set to 14m to avoid interfering with distant nodes. Packets can only be transmitted to the top, bottom, left or right node within the transmission range of the forwarding node as shown by nodes 6 and 14, but not diagonally. The network is designed with redundant links along the corridor where nodes are place in parallel to each other to form two possible paths between the source and the destination (Chipara et al., 2010). This alternative node allows individual sensor node to send the packet via the alternative path when the next hop node failed. In a traditional network, it is a common to provide an additional backup link next to the current one to provide redundancy. With this additional link, a route can be established quickly without traversing back to the direction of the source and can reduce the network recovery time.

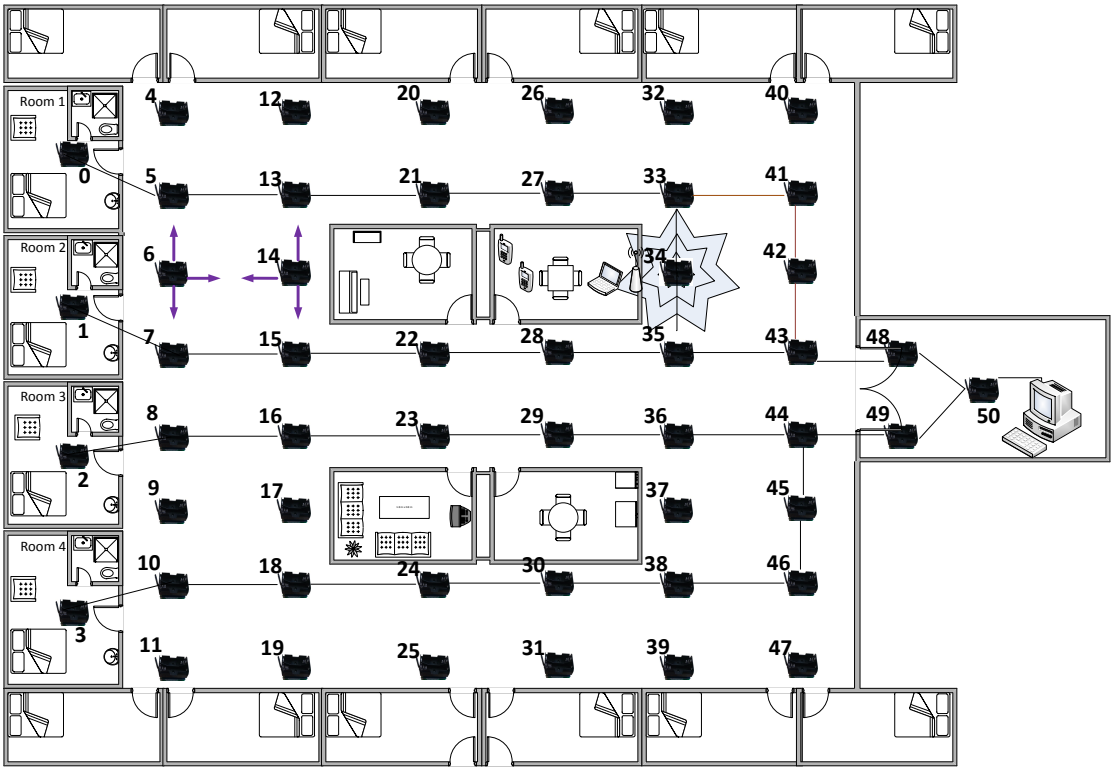



Figure 4.3: Network topology based on the indoor deployment for critical health monitoring networks

### 4.3.3 Simulation Parameters

Table 4.1 provides the simulation parameters adopted for this experiment. Extensive simulations are performed using the ns-2.34 simulator (NS2, 2002). The MAC layer is configured to operate in CSMA/CA mode as defined in the IEEE 802.15.4 (Kessler and Shepard, 1997) since it is commonly used in real sensor platform like

TinyOS. The two-ray ground propagation model was applied as it is deterministic and will not create additional packet drop (Zamalloa and Krishnamachari, 2007). The node can either receive the packet or not depending on whether it is within the receiving range of the transmitting node. Hence, this transmission model will not affect the failure model used in the experiment. A CBR packet was generated to model the periodic patient data that is required by the application. Similar to the application proposed by (Chipara et al., 2010), packet data is sampled and sent every one second to avoid any loss of vital data. In each experiment, the simulation for each of the routing protocols (MRP, NST and AODV) is run for 120 simulation seconds. Due to the stochastic nature of the simulator, each experiment was repeated $N = 50$ to reduce the uncertainty.

Table 4.1: NS-2 Parameters to evaluate MRP.

| **Parameters** | **Values** |
|---|---|
| *Simulation area:* | 200x200m |
| *Number of nodes* | 51 nodes |
| *Initial Energy* | 30J |
| *Transmission interval:* | 1 s |
| *Propagation model:* | Two-Ray Ground |
| *MAC*: | 802.15.4 (CSMA/CA) with LLN enabled |
| *Routing Protocol:* | AODV, NST-AODV, MRP-AODV [1] |

### 4.3.4 Simulation of failures

To investigate how irregular failures can affect the behaviour of the MRP and the network performance, different failures are injected to the nodes. Six failure durations of 0.5s, 2s, 5s, 10s, 20s and permanent are arbitrary chosen to represent different transient failures of short to long temporal behaviours. These errors are induced after the packet has traversed more than $H_t/2$ hops to allow the possibility of the LD to be activated where $H_t$ is the maximum number of hops from the source to the destination. Following the ON/OFF failure model specified in Section 3.1.2, an active node receiving a packet with hop count equal to $H_t/2$ is failed. We also vary the number of failing nodes as more than one sensor nodes may be affected spatially by the interfering signal in real world.

### 4.3.5 Results

To evaluate the difference between the performance of the routing algorithms, we compute the performance metrics defined in Section 3.1.5, namely PDR, average end to end delay, average energy consumed by the node, and routing overhead for different number of failing nodes with different durations. As all the results from different numbers of the failures have shown similar trends, the data taken from simulation with 10 failing nodes are selected and presented as box-whiskers plot for comparison in Figure 4.4 and 4.5. The use of box-whiskers provides a clear understanding on the distribution of the results collected from the simulation and it shows the maximum and minimum values of each of the performance metrics. Table 4.3 shows the numerical median and mean values of each routing protocol for detailed comparison.

From the box-whiskers plot in Figure 4.4(a), all the three routing protocols exhibit statistically different PDRs as computed in Table 4.4. The number of packets delivered by MRP are higher than the NST and AODV. When no error is injected, Figure 4.4(a) shows that the PDR of AODV (Table 4.3: median=92.04%, mean=89.77%) is significantly less than NST (Table 4.3: median=97.15%, mean=96.16%, Table 4.4: ranksum $p$-value=0.01, Vargha-Delaney $A$-value=0.75) and MRP (Table 4.3: median=98.01%, mean=97.56%, Table: 4.4: ranksum $p$-value=4.40e-07, Vargha-Delaney $A$-value=0.85).

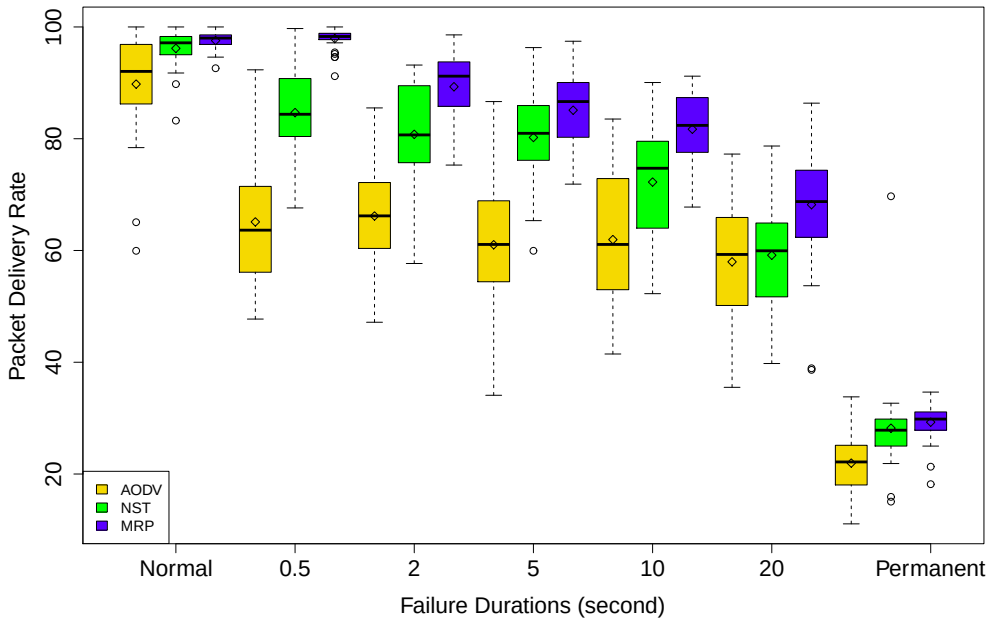
Further analysis on the simulator logs reveals that the data packets dropped in AODV are related to the Link Quality Issue (in **bold**).

Table 4.2: NS-2 traces showing the data packet (CBR) dropped (D) at the MAC layer (MAC) due to collisions (LQI) (In **bold**).

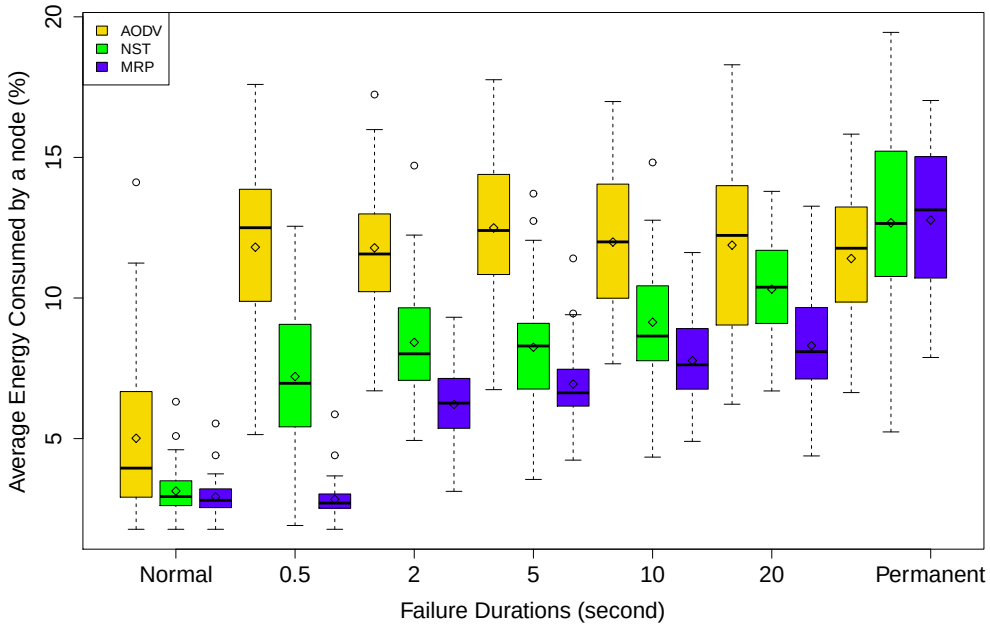
| action | time | node | at | why | pktid | pkt | |
|---|---|---|---|---|---|---|---|
| *D* | *92.5965056* | *11* | *MAC* | *LQI* | *0* | *AODV* | *39* |
| *D* | *92.596505613* | *18* | *MAC* | *LQI* | *0* | *AODV* | *39* |
| **D** | **92.6020416** | **11** | **MAC** | **LQI** | **780** | **cbr** | **107** |
| **D** | **92.605547722** | **27** | **MAC** | **LQI** | **775** | **cbr** | **107** |
| **D** | **92.6071616** | **11** | **MAC** | **LQI** | **780** | **cbr** | **107** |
| **D** | **92.6132416** | **11** | **MAC** | **LQI** | **780** | **cbr** | **107** |
| **D** | **92.633721594** | **10** | **MAC** | **LQI** | **795** | **cbr** | **107** |
| *D* | *92.644395627* | *11* | *MAC* | *LQI* | *0* | *AODV* | *39* |

From the log shown below, the transmission in node 11 and 10 have experienced error in the MAC layer (MAC) where data packets (CBR) have been dropped (D) due to packet collision (LQI).

Due to the random access of the channel in CSMA, higher numbers of packet drop in AODV have occurred in the network resulting more requests for route

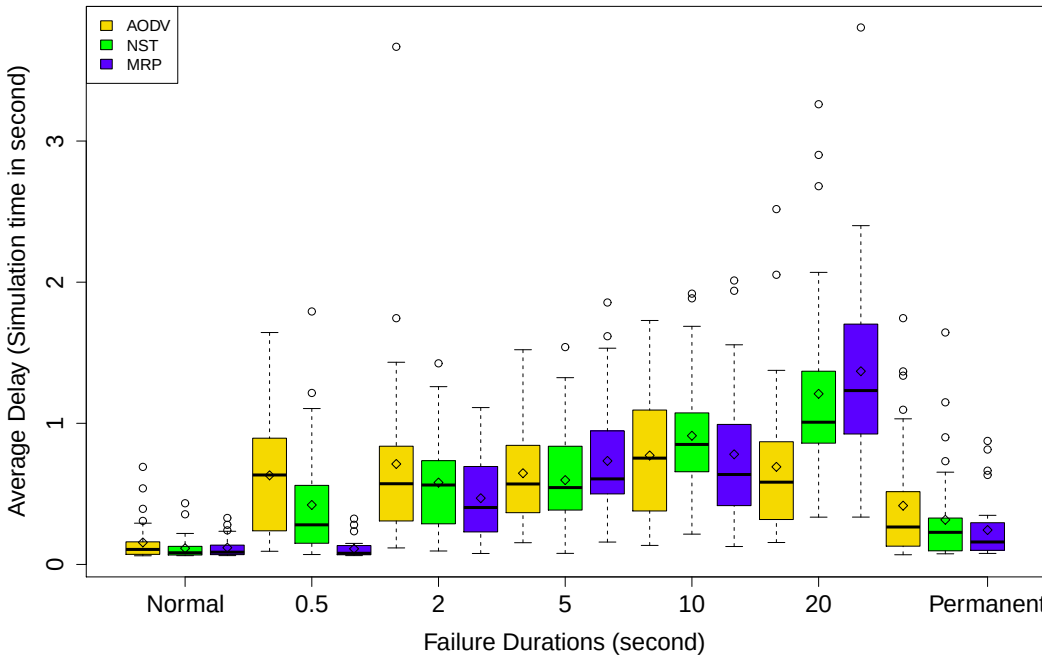



(a) Higher PDR is observed in MRP across all failures.

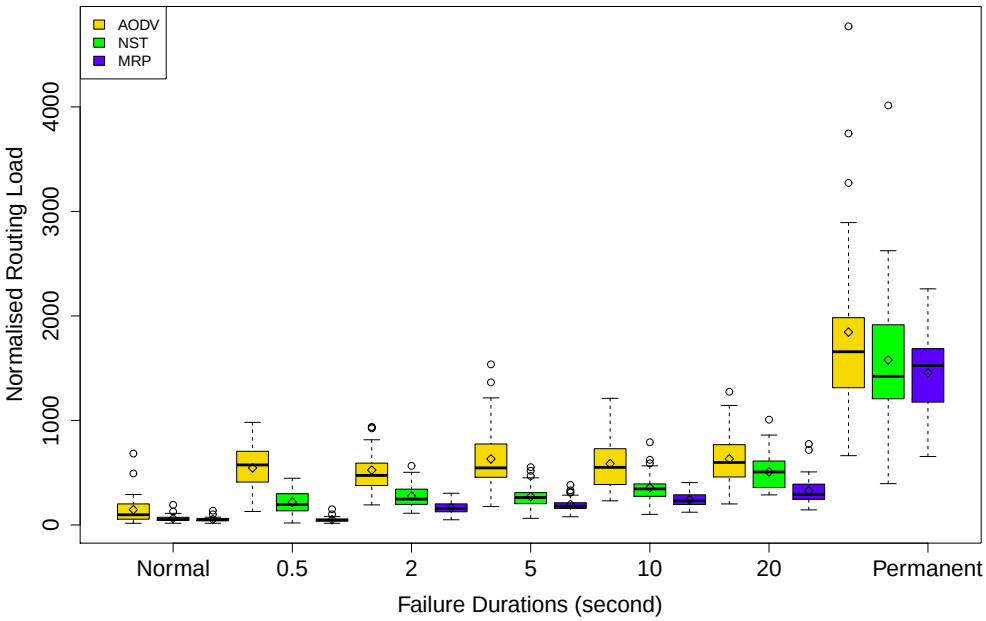



(b) Higher energy is consumed in AODV due to the transmission of routing packets.

Figure 4.4: Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for PDR and Energy Consumption on the performance of MRP against AODV and NST




(a) The average delay for all the routing protocols is similar when the failure duration increases.




(b) More routing packets are generated without the retransmission in MRP and NST.

Figure 4.5: Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for Average Delay and Routing Overhead on the performance of MRP against AODV and NST

Table 4.3: The performance results in term of PDR, average energy remaining in a node, routing packet required to deliver a packet, and the average end to end delay for the three protocols: MRP, NST and AODV. Shown are the median and mean over the 50 runs for different failure durations of 0.5s, 2s, 5s, 10s, 20s exposed to 10 failing nodes.

| | Failure | Median | | | Mean | | |
|---|---|---|---|---|---|---|---|
| | Duration (s) | AODV | NST | MRP | AODV | NST | MRP |
| PDR (%) | 0 | 92.04 | 97.15 | **98.01** | 89.77 | 96.16 | **97.56** |
| | 0.5 | 69.60 | 89.77 | **98.29** | 70.66 | 88.40 | **97.10** |
| | 2 | 72.44 | 84.94 | **90.62** | 70.34 | 84.69 | **89.72** |
| | 5 | 65.62 | 85.51 | **88.92** | 66.98 | 83.83 | **88.22** |
| | 10 | 72.44 | 78.69 | **83.23** | 70.19 | 78.36 | **81.93** |
| | 20 | 68.75 | 67.61 | **74.43** | 67.42 | 66.69 | **72.95** |
| | Permanent | 23.29 | 27.55 | **29.54** | 22.11 | 27.84 | **28.92** |
| Average Energy Consumption (%) | 0 | 3.95 | 2.94 | **2.81** | 5.01 | 3.14 | **2.93** |
| | 0.5 | 12.50 | 6.97 | **2.71** | 11.81 | 7.21 | **2.84** |
| | 2 | 11.57 | 8.02 | **6.26** | 11.79 | 8.42 | **6.22** |
| | 5 | 12.40 | 8.29 | **6.63** | 12.50 | 8.25 | **6.94** |
| | 10 | 12.00 | **8.65** | 7.62 | 11.99 | 9.14 | **7.77** |
| | 20 | 12.23 | 10.39 | **8.09** | 11.88 | 10.32 | **8.30** |
| | Permanent | 11.77 | **12.65** | 13.13 | 11.40 | 12.68 | **12.78** |
| Routing Load | 0 | 97 | 52 | **49** | 146 | 62 | **53** |
| | 0.5 | 445 | 156 | **49** | 420 | 167 | **55** |
| | 2 | 335 | 188 | **155** | 438 | 209 | **146** |
| | 5 | 486 | 189 | **171** | 528 | 207 | **165** |
| | 10 | 376 | 267 | **216** | 453 | 272 | **216** |
| | 20 | 396 | 334 | **263** | 461 | 362 | **267** |
| | Permanent | 1618 | 1692 | **1459** | 1802 | 1802 | **1481** |
| DLY (s) | 0 | 0.10 | **0.08** | **0.08** | 0.15 | **0.11** | **0.11** |
| | 0.5 | 0.49 | 0.24 | **0.10** | 0.50 | 0.35 | **0.12** |
| | 2 | 0.45 | 0.40 | **0.35** | 0.53 | 0.43 | **0.40** |
| | 5 | 0.43 | **0.36** | 0.57 | **0.51** | 0.52 | 0.61 |
| | 10 | **0.32** | 0.75 | 0.63 | **0.40** | 0.82 | 0.76 |
| | 20 | **0.44** | 0.77 | 1.28 | **0.56** | 0.94 | 1.37 |
| | Permanent | 0.31 | 0.24 | **0.18** | 0.49 | 0.36 | **0.28** |

discovery and higher routing packet as depicted in Figure 4.5(b). In Table 4.3. AODV (97 packets (median) and 146 packets (mean)) produces more routing overhead compared to NST (52 packets (median) and 62 packets (mean)) and (49 packets (median) and 53 packets (mean)). We believe the additional number of routing packets generated has lowered the number of PDRs in AODV observed in Figure 4.4(a).

As the failure duration increases from 0.5s to 20s, the PDRs of NST and MRP decrease while AODV maintains a fluctuating median of 69.60% to 72.44% (mean:

Table 4.4: *p*- and *A*-values indicating the difference between MRP and AODV, MRP and NST, and NST and AODV for different failure durations. Numbers highlighted in ***bold-italic*** are significant, comprising a large effect

| **Significance Tests** | | **P-Test (*p*-value)** | | | **A-test (*A*-value)** | | |
|---|---|---|---|---|---|---|---|
| **Metrics** | **Failure (Nodes)** | **MRP AODV** | **MRP NST** | **NST AODV** | **MRP AODV** | **MRP NST** | **NST AODV** |
| *PDR* | 0 | ***4.40e-07*** | ***0.08*** | ***0.01*** | ***0.85*** | 0.62 | ***0.75*** |
| | 0.5 | ***6.83e-13*** | ***3.93e-11*** | ***4.57e-09*** | ***1.00*** | ***0.96*** | ***0.91*** |
| | 2 | ***5.35e-12*** | ***3.82e-05*** | ***2.50e-07*** | ***0.98*** | ***0.79*** | ***0.86*** |
| | 5 | ***1.82e-11*** | ***0.01*** | ***1.39e-09*** | ***0.97*** | ***0.71*** | ***0.92*** |
| | 10 | ***3.95e-10*** | ***0.01*** | ***0.01*** | ***0.94*** | ***0.77*** | ***0.74*** |
| | 20 | ***0.01*** | ***0.01*** | 0.66 | ***0.76*** | ***0.74*** | 0.53 |
| | $\infty$ | ***2.46e-08*** | ***0.02*** | ***3.69e-05*** | ***0.89*** | 0.66 | ***0.79*** |
| *Average Energy Remains* | 0 | ***9.86e-05*** | 0.52 | ***0.01*** | ***0.77*** | 0.55 | ***0.73*** |
| | 0.5 | ***3.56e-20*** | ***1.37e-14*** | ***4.98e-09*** | ***1.00*** | ***0.96*** | ***0.88*** |
| | 2 | ***4.83e-17*** | ***1.60e-06*** | ***3.21e-08*** | ***0.98*** | ***0.82*** | ***0.86*** |
| | 5 | ***2.72e-14*** | ***0.01*** | ***2.12e-09*** | ***0.96*** | ***0.71*** | ***0.88*** |
| | 10 | ***1.83e-11*** | ***0.01*** | ***7.84e-06*** | ***0.92*** | ***0.71*** | ***0.80*** |
| | 20 | ***9.81e-07*** | ***9.11e-05*** | ***0.02*** | ***0.82*** | ***0.76*** | 0.66 |
| | $\infty$ | ***0.02*** | 0.89 | 0.07 | 0.66 | 0.51 | 0.63 |
| *Routing Load* | 0 | ***5.46e-05*** | 0.41 | ***0.01*** | ***0.78*** | 0.56 | ***0.74*** |
| | 0.5 | ***3.56e-20*** | ***1.37e-14*** | ***4.25e-10*** | ***1.00*** | ***0.96*** | ***0.90*** |
| | 2 | ***2.16e-17*** | ***1.13e-06*** | ***6.00e-09*** | ***0.99*** | ***0.82*** | ***0.88*** |
| | 5 | ***1.44e-15*** | ***0.01*** | ***2.25e-10*** | ***0.97*** | ***0.75*** | ***0.90*** |
| | 10 | ***1.85e-13*** | ***3.53e-05*** | ***1.37e-05*** | ***0.95*** | ***0.78*** | ***0.79*** |
| | 20 | ***2.57e-09*** | ***2.25e-06*** | ***0.02*** | ***0.88*** | ***0.81*** | 0.66 |
| | $\infty$ | 0.06 | 0.62 | 0.26 | 0.63 | 0.54 | 0.58 |
| *Average Delay* | 0 | 0.39 | 0.28 | 0.11 | 0.56 | 0.58 | 0.61 |
| | 0.5 | ***2.10e-12*** | ***2.28e-08*** | ***0.02*** | ***0.93*** | ***0.86*** | 0.66 |
| | 2 | 0.08 | 0.23 | 0.57 | 0.62 | 0.58 | 0.54 |
| | 5 | 0.34 | 0.16 | 0.67 | 0.57 | 0.60 | 0.53 |
| | 10 | 0.97 | 0.11 | 0.17 | 0.50 | 0.61 | 0.60 |
| | 20 | ***5.11e-07*** | 0.12 | ***2.97e-05*** | ***0.83*** | 0.61 | ***0.78*** |
| | $\infty$ | 0.06 | 0.67 | 0.20 | 0.63 | 0.53 | 0.59 |

66.98 to 70.66%) in Figure 4.4(a). The PDR of MRP degrades slower than NST as the failure duration is increased. In Table 4.3, the PDR of MRP decreases from 98.29% to 74.43% (mean: 88.40% to 66.69%) while NST decreases from 89.77% to 67.61% (mean: 88.40 to 66.69). The PDR of MRP is significantly better than NST and AODV (Table 4.4: ranksum $p$-value $< 0.01$ and Vargha-Delaney $A$-value $> 0.71$) when errors are injected into the nodes. When the failure duration is increased to 20s, there is no significance different between the PDR of NST (mean=66.69%, median=67.61%) and AODV (mean=67.42%, median=67.61%). The same PDR is observed in AODV when errors are injected regardless of the duration of the failure.

When no error is injected, higher energy consumption is also observed in Figure 4.4(b) for AODV (4.3: median=3.95%, mean=5.01%) compare to NST (4.3: median=2.94%, mean=3.14%) and MRP (4.3: median=2.81, mean=2.93). As a transient error of 0.5s is injected to the network, the energy consumption of AODV increases from 3.95% to 12.5% (median) and 5.01% to 11.81% (mean) compared to 2.94% to 6.97% (median) and 3.14% to 7.21% (median) in NST. MRP has the lowest energy consumption with 2.71% (median) and 2.84% (mean). As the failure duration increases from 2s to 20s, the rate of energy consumption of MRP shown in 4.3 is slower than NST while the energy consumption of AODV remains consistent between 11.57% to 12.50% (median) and 11.79% to 12.50% (mean). The statistical tests in Table 4.4 also show that the energy consumption of MRP is significantly lower than NST and AODV. The higher energy consumption observed in AODV and NST are caused by the additional transmission of the routing overhead. As longer failure durations are injected, more route discoveries are activated by the nodes and therefore more energy is consumed.

In terms of the packet end to end delay, MRP has a lowest end to end delay when the failure duration is less than 0.5s. while the packet latency in AODV is maintained between 0.32s to 0.43s. The end to end delay in AODV is consistent throughout the failure as packets that cannot be transmitted in AODV were dropped and was not accounted for during the calculation of the average delay. Hence, a lower delay is observed in AODV even at high failure duration. In MRP, the packets are stored the routing queue while it tries to select the routing protocol to make any attempt to deliver the packet. Although this buffering process has introduced additional delay in MRP as shown in Figure 4.5(a) that is smaller than NST and AODV, results from the statistical tests in Table 4.4 have shown that the differences between the delay are not statistically significant.

#### Discussion on the effect of failure duration

In this analysis, we have shown that MRP can achieve better performance than NST and AODV. From the results, higher PDR has been observed in MRP than NST and AODV in figure 4.4(a). The built-in switching mechanism and the localised decision whether to activate local repair, retransmission or drop the packet have allowed individual nodes to reduce the number of routing packets. This reduction in number of routing packets can be seen in Figure 4.5(b) where MRP has the lowest routing overhead required to ensure packet delivery. MRP has also reduced the amount of packet transmission leading to the reduction of the energy consumption in the node. The statistical tests have verified that differences in the PDR, energy consumption and routing overhead are both statistical and scientifically significant. Hence the MRP can tolerate failures with different durations and is more reliable and energy efficient than NST and AODV.

## 4.4 On Robustness: 2. The effects of varying the number of failing nodes

As the performance of the WSNs can be affected by interference spatially, in this section we investigate the effects of varying the number of nodes on the performance of MRP, NST and AODV.

### 4.4.1 Results

We use the data obtained from Section 4.3 to compute the PDR, energy consumption, routing overhead and average delay for different numbers of failure up to the maximum 14 nodes before a total network failure occurs. We discuss each of the results separately in a tabulate format in the following subsections.

#### Packet Delivery Rate

Table 4.5 enumerates the median PDR achieved by each of the routing protocol when the number of failing nodes induced with failures increases. The results show that the PDR achieved by MRP is significantly higher than AODV and NST with 0.5s failure duration ($p$-value $< 0.01$ and $A$-value $< 0.73$). When errors with 0.5s are introduced in the nodes, the PDR of AODV degrades faster than NST and AODV. When the number of failures with 0.5s duration is increased from 1 to the maximum (14) nodes, the PDR of AODV decreases from 88.35% to 63.64% (by 24.71%) and NST decreases from 94.89% to 84.38% (by 10.51%). The PDR of MRP

remains consistent ($98.01\% < \text{PDR} < 98.5\%$) as the number of failures increases at 0.5s. The ability of the nodes to avoid local route discovery has reduced the number of routing packets generated as shown in Table 4.7 and reduces the probability of packet collisions. Visual inspection on the simulation traces also reveal that the number of packets drop related to LQI is also less in MRP compared to AODV. This shows that the MRP is more tolerate to failure with 0.5% duration.

When failures with longer duration (2s, 10s and 20s) is introduced, the PDR of MRPS begin to decrease slowly (by 10% at 2s, 16% at 10s and 27.56% at 20s) Table 4.5. However, the rate of the PDR degrades in MRP is smaller than NST and AODV. The results also show that the PDR of AODV is not affected by the failure duration as the PDR generated by each number of failures is similar for all failure duration. Statistical test have shown that the differences in PDR of AODV between different durations are not significant.

We also evaluate the distribution of the results between the PDR of MRP and AODV, MRP and NST, and NST and AODV using non-parametric statistical tests to determine significance of the PDR difference for 5s, 2s, 10s and 20s failure durations. The minimum $p$-values and maximum $A$-value between MRP and AODV, MRP and NST, and NST and AODV are plotted against the number of failures in Figure 4.6, 4.7, 4.8 and 4.9 respectively. The graph in 4.6 shows that the differences between MRP and AODV are statistical significant at 95% confidence interval when the failure durations is 0.5s. Between MRP and NST (Downward-pointing triangle), most of the $p$-values are below the 0.05 significance level as shown by the dotted line in Figure 4.6 and 4.7. There are less than 4 cases when the $p$-values are more than 0.05 for each failure duration. The difference of between MRP and AODV are also scientifically significance as they exhibit a large effect size with the $A$-values are greater than 0.71 in Figure 4.8 and 4.9. We can conclude that MRP deliver more packets than NST and AODV and the performance is significantly better than AODV.

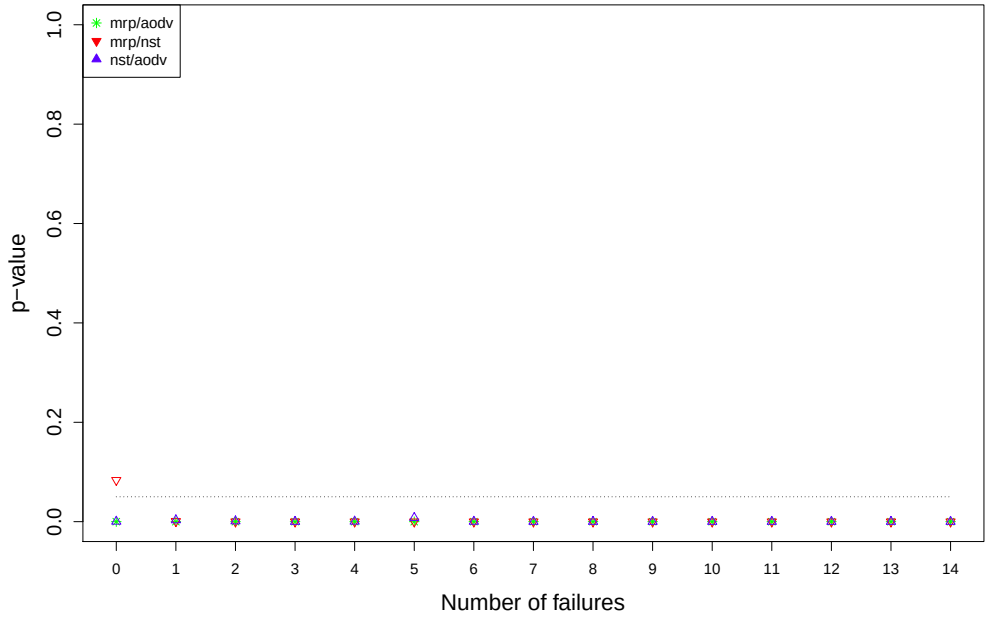
**Average Energy**

In the previous section, we have evaluated how the number of failures can reduce the packet delivery of AODV and NST. In this section, we analyse the energy efficiency of the routing protocols. We depict the median taken from 50 different seeded runs for each failure in Table 4.6. To better understand the significant difference between the energy consumptions, the $p$- and $A$-values are computed statistically and presented graphically in Figure 4.10, 4.11, 4.12 and 4.13

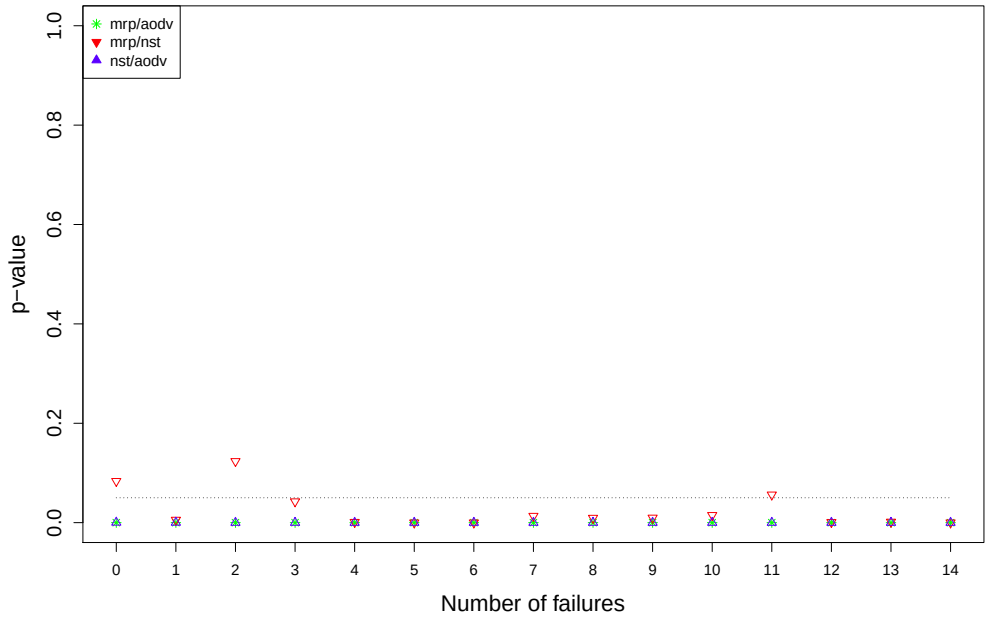
During normal network operation, both MRP and NST only consume minimal energy of 2.81% and 2.94% respectively compare to AODV of 3.95% as shown

Table 4.5: The median of the PDR calculated from 50 runs for different number of failures injected with different failure duration for MRP, NST and AODV.

| **Number of Failures** | **PDR (%) for 0.5s** | | | **PDR (%) for 2s** | | |
|---|---|---|---|---|---|---|
| | AODV | NST | MRP | AODV | NST | MRP |
| 0 | 92.05 | 97.16 | **98.01** | 92.05 | 97.16 | **98.01** |
| 1 | 88.35 | 94.89 | **98.30** | 88.35 | 94.60 | **97.44** |
| 2 | 88.35 | 93.75 | **98.30** | 78.69 | 94.60 | **96.59** |
| 3 | 78.69 | 92.61 | **98.58** | 83.52 | 93.47 | **97.16** |
| 4 | 81.82 | 92.05 | **98.30** | 77.56 | 91.76 | **96.59** |
| 5 | 82.39 | 91.76 | **98.30** | 79.55 | 90.34 | **96.88** |
| 6 | 79.55 | 89.77 | **98.30** | 73.86 | 90.63 | **96.59** |
| 7 | 76.42 | 88.64 | **98.30** | 75.85 | 89.77 | **93.47** |
| 8 | 70.17 | 90.91 | **98.30** | 73.58 | 87.22 | **93.47** |
| 9 | 68.47 | 87.78 | **98.58** | 71.31 | 87.22 | **92.90** |
| 10 | 69.60 | 92.05 | **98.30** | 72.44 | 91.76 | **96.59** |
| 11 | 65.63 | 84.94 | **98.30** | 71.59 | 86.93 | **88.64** |
| 12 | 65.63 | 87.22 | **98.30** | 66.48 | 82.10 | **91.19** |
| 13 | 64.49 | 84.38 | **98.30** | 66.19 | 80.68 | **90.34** |
| 14 | 63.64 | 84.38 | **98.30** | 66.19 | 84.09 | **91.19** |
| **Number of Failures** | **PDR (%) for 10s** | | | **PDR (%) for 20s** | | |
| 1 | 87.22 | 94.89 | **97.44** | 89.49 | 95.45 | **97.44** |
| 2 | 83.81 | 93.47 | **96.59** | 83.24 | 90.34 | **96.31** |
| 3 | 81.82 | 90.91 | **96.31** | 82.95 | 91.19 | **96.31** |
| 4 | 84.66 | 89.20 | **96.31** | 73.58 | 80.11 | **90.91** |
| 5 | 77.56 | 86.65 | **92.33** | 73.58 | 71.88 | **91.48** |
| 6 | 74.43 | 86.08 | **92.33** | 75.28 | 75.28 | **84.66** |
| 7 | 72.16 | 87.22 | **87.78** | 70.74 | 75.00 | **84.66** |
| 8 | 76.70 | 85.23 | **86.08** | 71.59 | 73.86 | **80.11** |
| 9 | 72.44 | 81.25 | **87.22** | 69.03 | 68.47 | **77.84** |
| 10 | 72.44 | 89.20 | **96.31** | 68.75 | 80.11 | **90.91** |
| 11 | 61.65 | 78.69 | **82.95** | 61.65 | 67.61 | **69.89** |
| 12 | 61.08 | 77.84 | **84.66** | 57.67 | 66.19 | **68.75** |
| 13 | 65.34 | 74.72 | **80.40** | 55.68 | 59.94 | **71.02** |
| 14 | 61.08 | 73.30 | **82.39** | 59.29 | 66.19 | **68.75** |

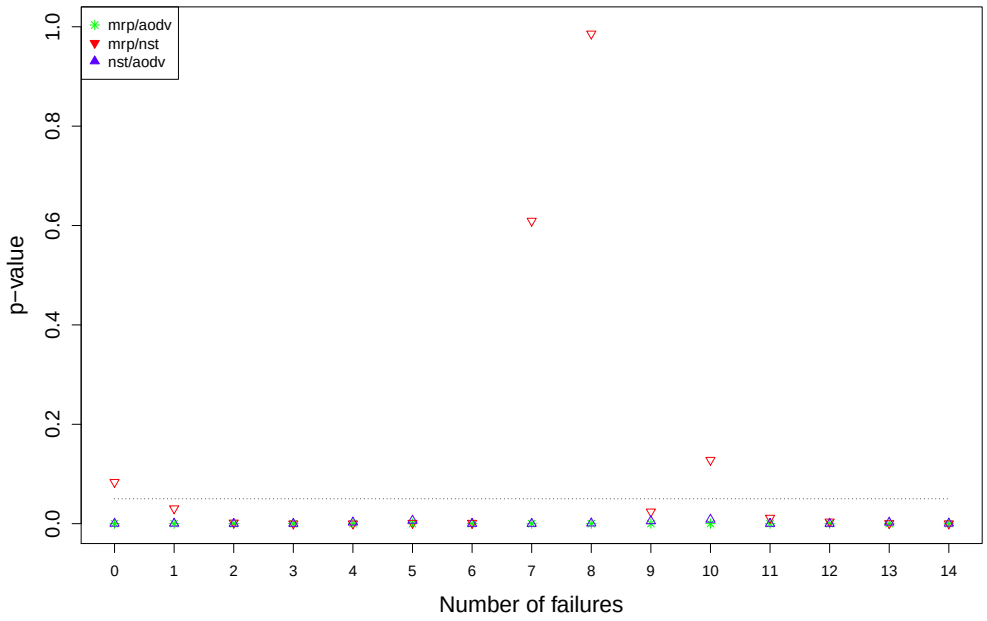



(a) The *p*-values for 0.5s

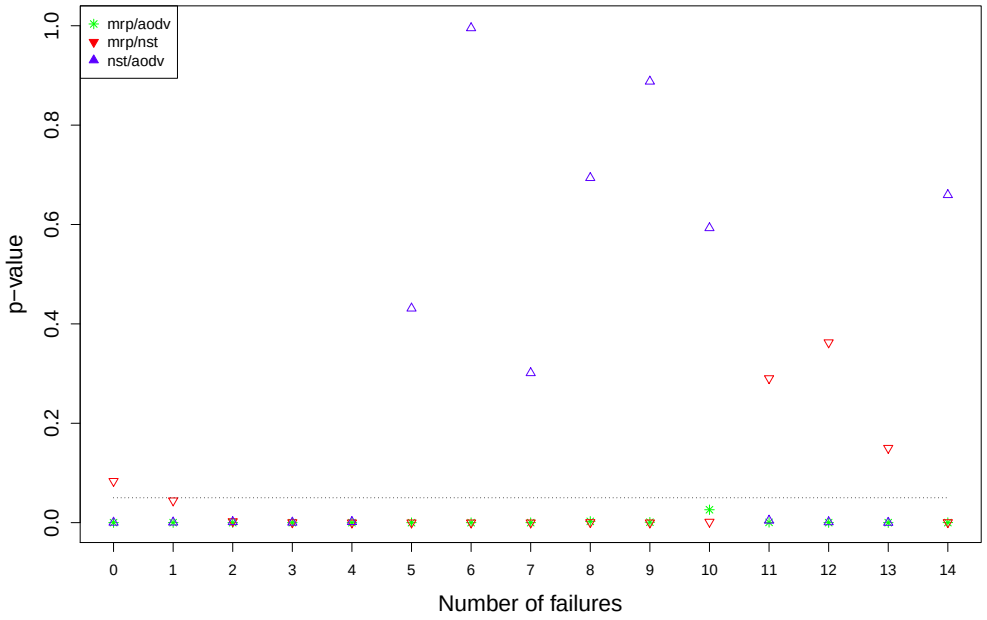



(b) The *p*-values for 2s

Figure 4.6: The *p*-values showing the statistical significant of PDR with 0.5s and 2s failure durations. The point below the line (at 0.05) shows that the results are significant.

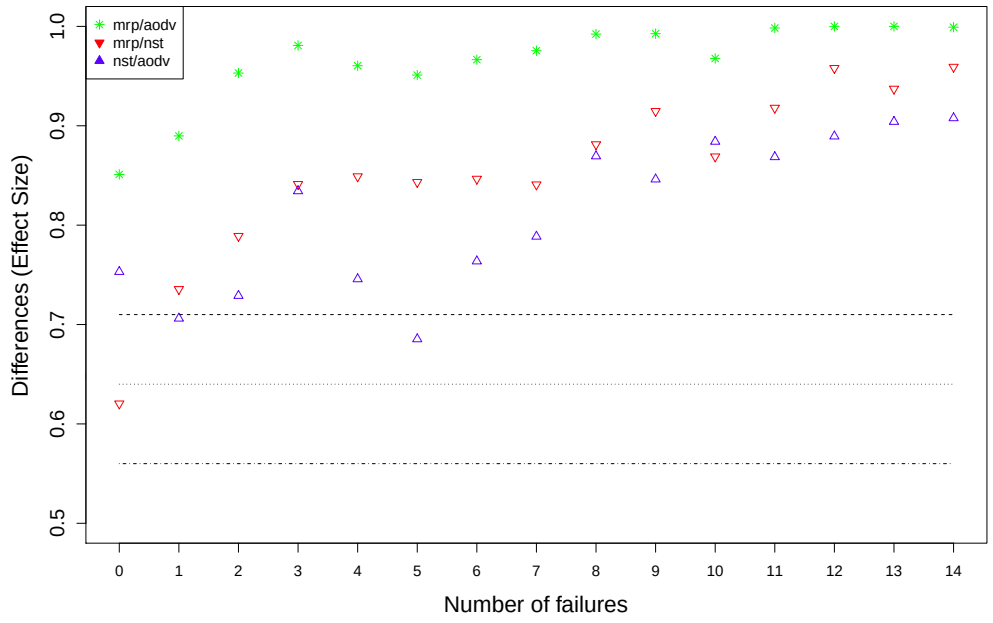



(a) The $p$-values for 10s

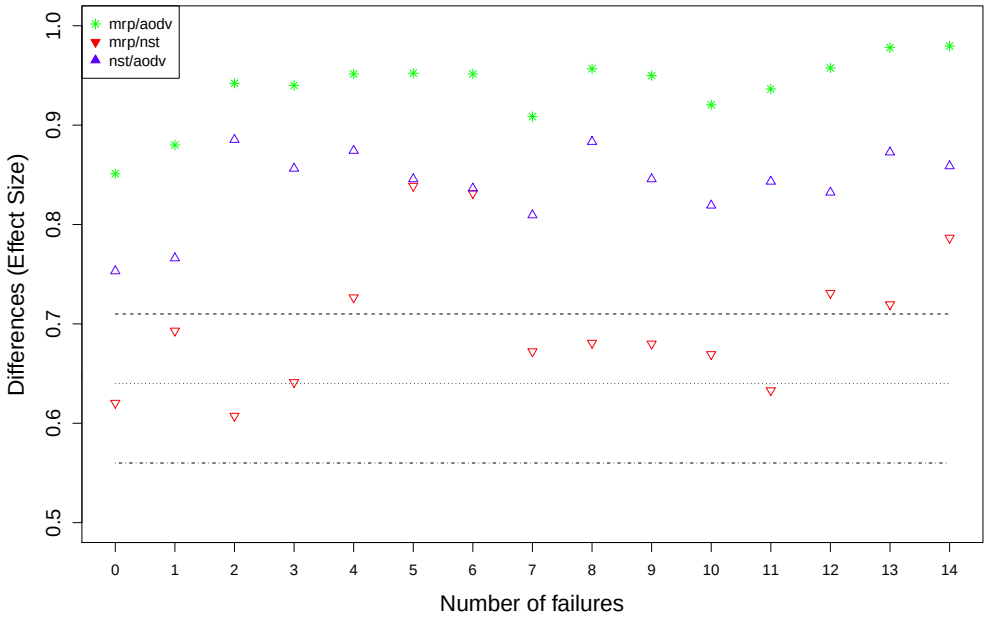



(b) The $p$-values for 20s

Figure 4.7: The $p$-values showing the statistical significant of PDR with 10s and 20s failure durations. The point below the line (at 0.05) shows that the results are significant.

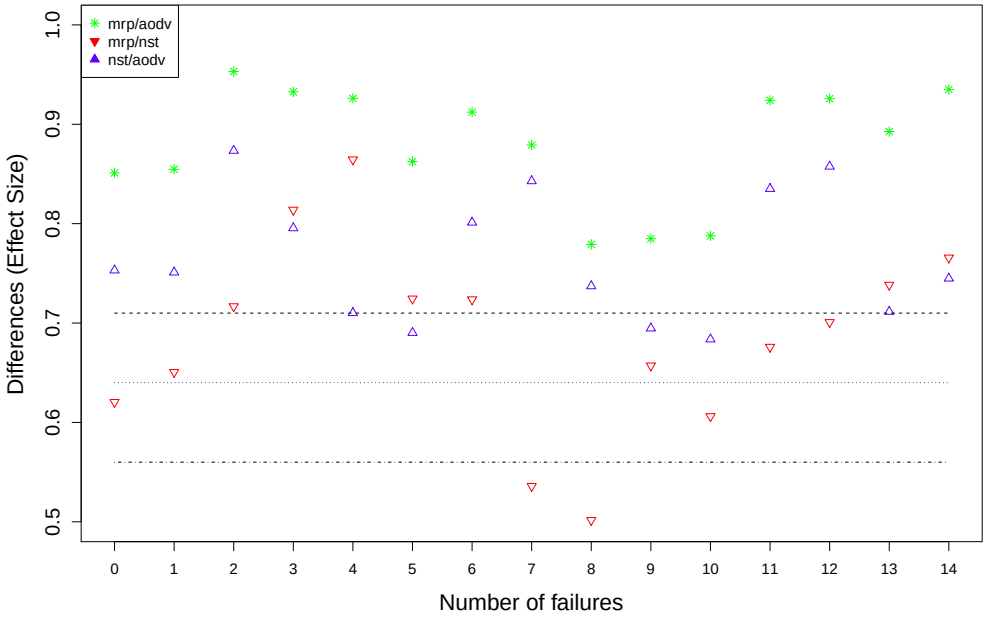



(a) The $A$-values for 0.5s

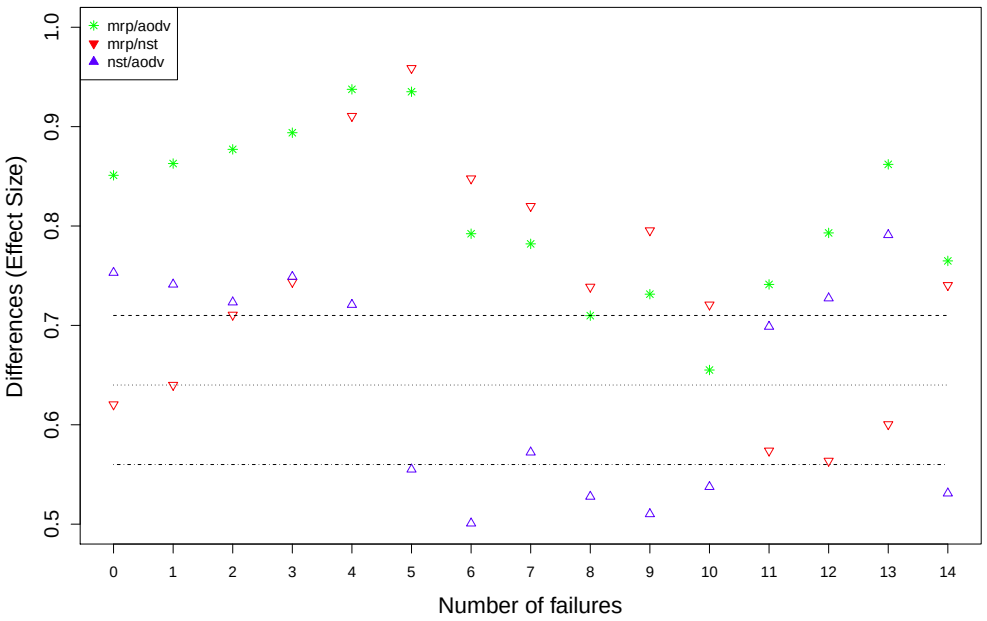



(b) The $A$-values for 2s

Figure 4.8: The $A$-values showing the scientific significant of PDR for with 0.5 and 2s failure durations. The points above the dashed line (at 0.71) show that the results are scientifically significant with large effect size.




(a) The $A$-values for 10s

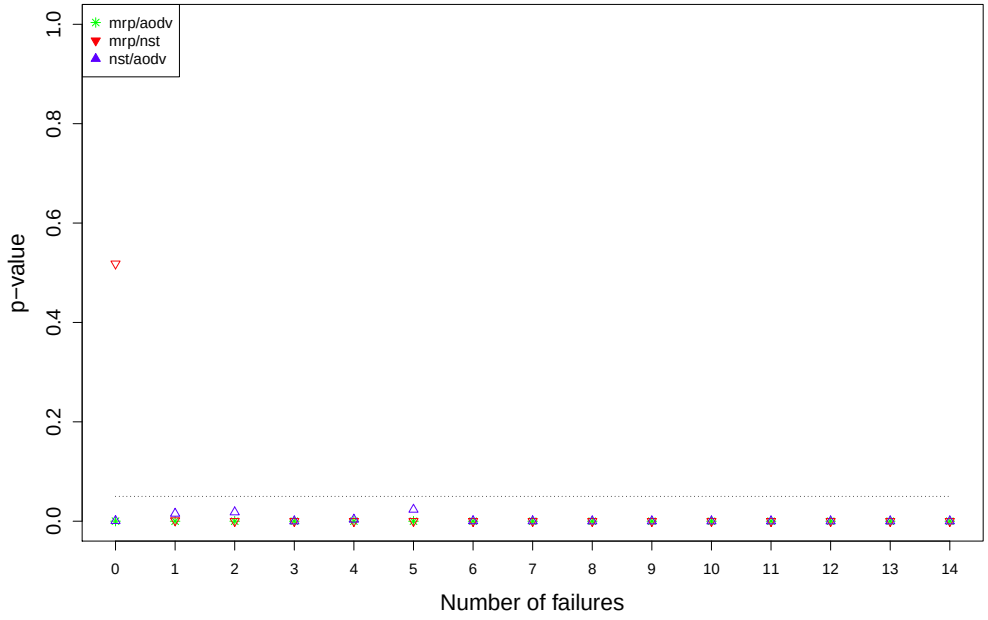



(b) The $A$-values for 20s

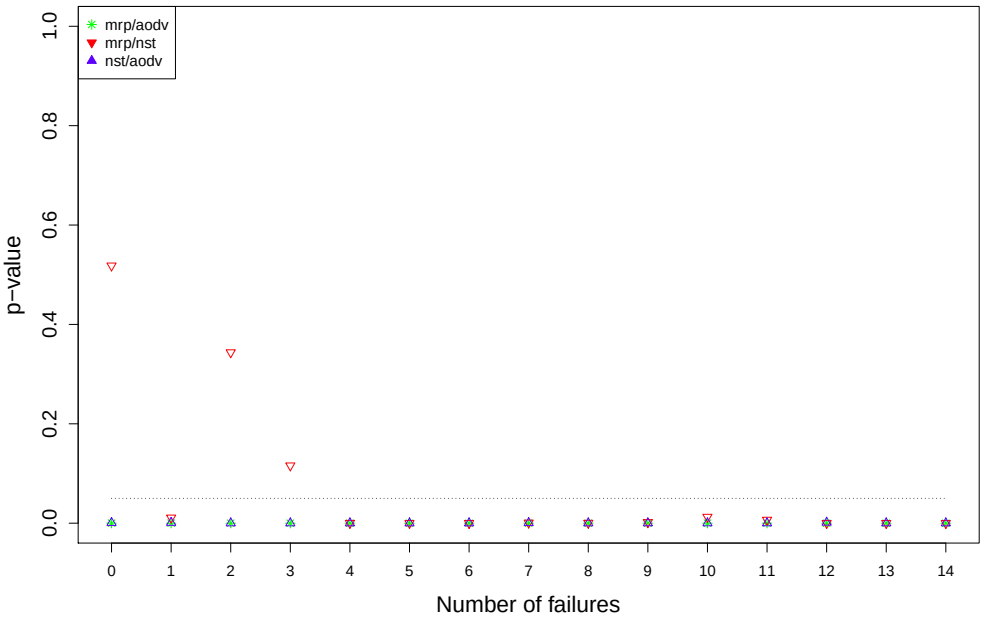
Figure 4.9: The $A$-values showing the scientific significant of PDR with 10s and 20s failure durations

Table 4.6: The median of the average energy remaining in a node for different number of failures injected with different failure duration for MRP, NST and AODV. The energy consumption increases as the number of failures introduced increases except for MRP with 0.5s failure durations.

| **Number of Failures** | **ENG (%) for 0.5s** | | | **ENG (%) for 2s** | | |
|---|---|---|---|---|---|---|
| | AODV | NST | MRP | AODV | NST | MRP |
| 0 | 3.95 | 2.94 | 2.81 | 3.95 | 2.94 | 2.81 |
| 1 | 5.26 | 4.05 | 2.77 | 5.27 | 3.75 | 2.94 |
| 2 | 5.31 | 4.05 | 2.73 | 7.58 | 3.75 | 3.55 |
| 3 | 7.72 | 4.51 | 2.71 | 6.66 | 4.12 | 3.37 |
| 4 | 6.58 | 5.02 | 2.76 | 8.28 | 5.41 | 3.63 |
| 5 | 6.58 | 5.35 | 2.73 | 8.25 | 5.62 | 3.50 |
| 6 | 7.56 | 5.11 | 2.83 | 8.91 | 5.44 | 3.95 |
| 7 | 7.72 | 5.60 | 2.73 | 8.21 | 6.31 | 5.13 |
| 8 | 9.68 | 5.25 | 2.76 | 9.20 | 6.43 | 5.18 |
| 9 | 9.90 | 5.43 | 2.73 | 8.82 | 6.81 | 5.18 |
| 10 | 10.89 | 5.02 | 2.76 | 8.85 | 5.41 | 3.63 |
| 11 | 11.03 | 6.34 | 2.73 | 10.11 | 7.13 | 5.68 |
| 12 | 11.56 | 6.97 | 2.73 | 10.11 | 7.68 | 5.72 |
| 13 | 11.46 | 6.97 | 2.74 | 11.14 | 8.02 | 5.92 |
| 14 | 12.50 | 7.00 | 2.71 | 11.57 | 7.91 | 6.26 |
| **Number of Failures** | **ENG (%) for 10s** | | | **ENG (%) for 20s** | | |
| 0 | 3.95 | 2.94 | 2.81 | 3.95 | 2.94 | 2.81 |
| 1 | 5.42 | 3.75 | 3.09 | 5.26 | 3.74 | 3.11 |
| 2 | 6.65 | 4.34 | 3.31 | 7.01 | 4.44 | 3.48 |
| 3 | 7.06 | 5.36 | 3.56 | 6.72 | 4.98 | 3.48 |
| 4 | 6.87 | 5.52 | 3.62 | 8.95 | 7.26 | 4.77 |
| 5 | 8.22 | 6.20 | 4.72 | 9.24 | 7.72 | 4.69 |
| 6 | 8.73 | 6.26 | 4.79 | 8.82 | 7.69 | 5.28 |
| 7 | 9.35 | 5.58 | 5.33 | 9.81 | 7.69 | 5.44 |
| 8 | 8.73 | 5.88 | 6.31 | 8.99 | 7.88 | 5.80 |
| 9 | 9.01 | 7.50 | 6.76 | 9.57 | 9.38 | 7.11 |
| 10 | 10.28 | 5.52 | 3.62 | 10.11 | 7.26 | 4.77 |
| 11 | 12.14 | 8.14 | 7.15 | 12.33 | 8.96 | 7.58 |
| 12 | 11.91 | 8.71 | 7.06 | 12.87 | 9.70 | 7.86 |
| 13 | 11.32 | 8.65 | 7.62 | 12.87 | 10.39 | 8.69 |
| 14 | 12.00 | 9.30 | 7.62 | 12.23 | 8.94 | 8.09 |

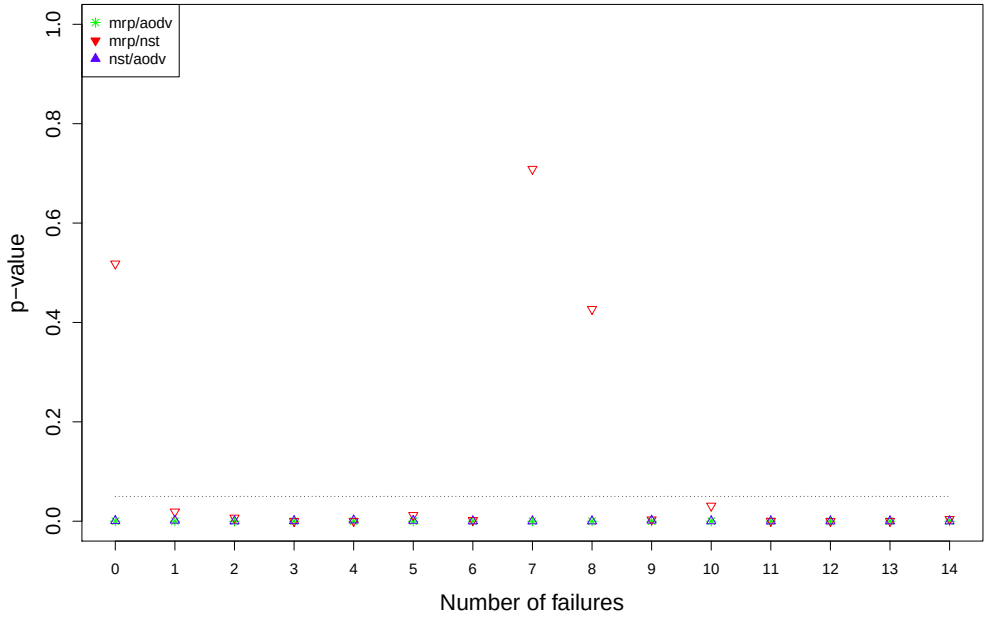



(a) The $p$-values for 0.5s

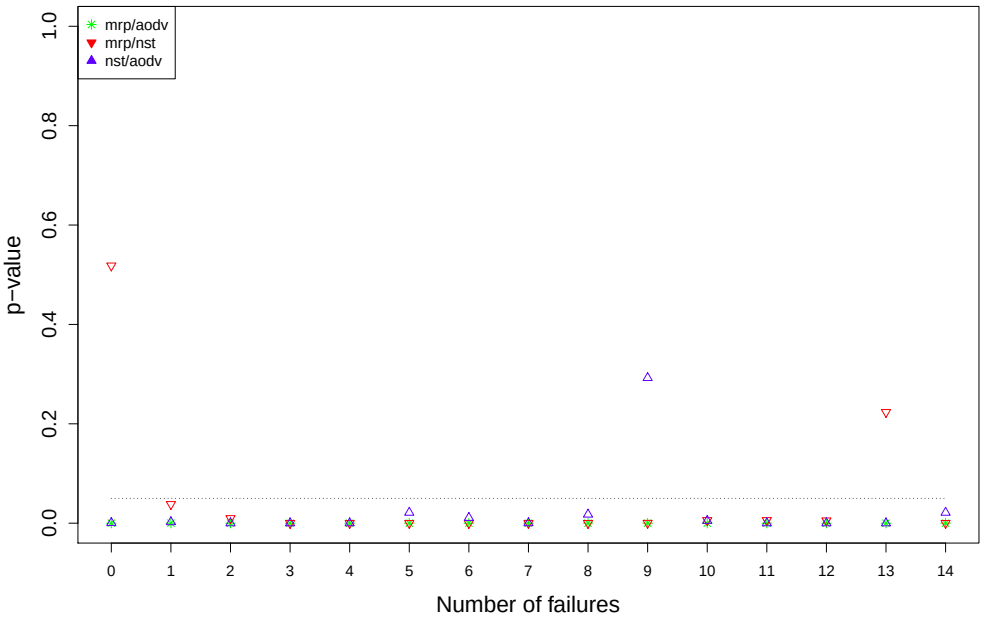



(b) The $p$-values for 2s

Figure 4.10: The $p$-values showing the statistical significant of energy difference.

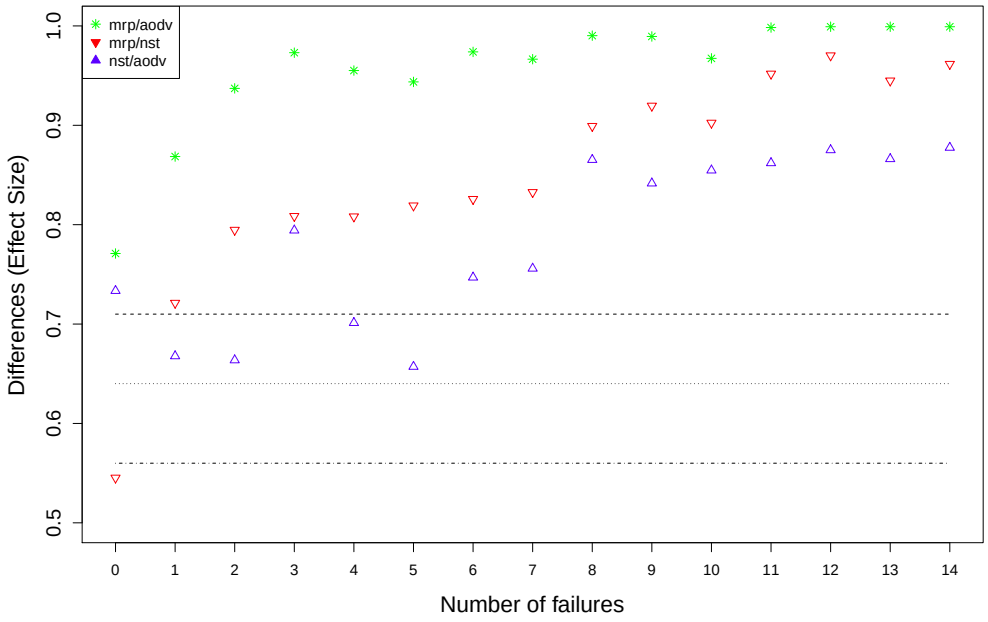



(a) The *p*-values for 10s

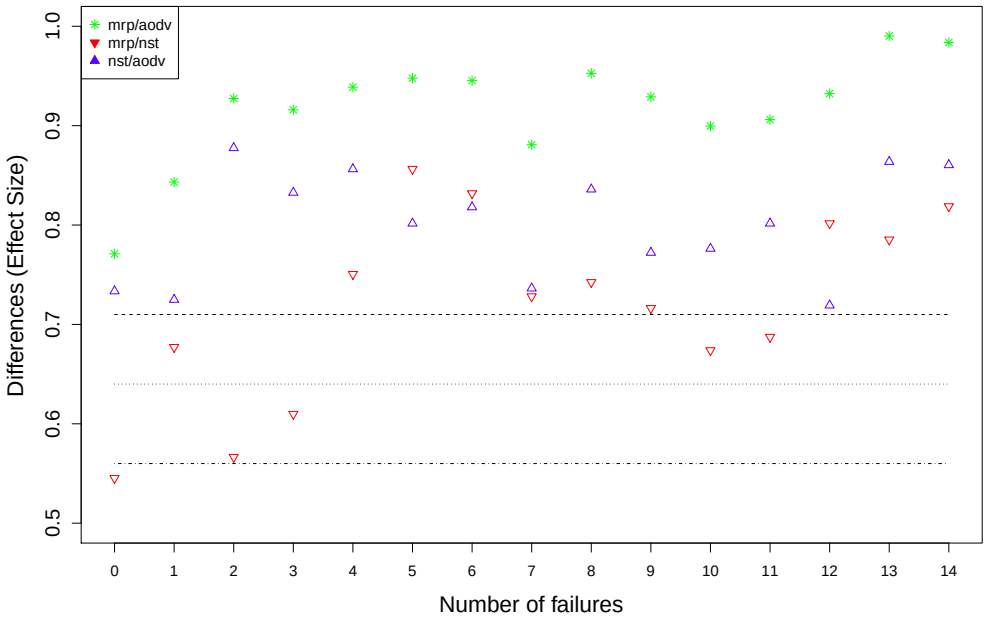



(b) The *p*-values for 20s

Figure 4.11: The *p*-values showing the statistical significant of energy difference.

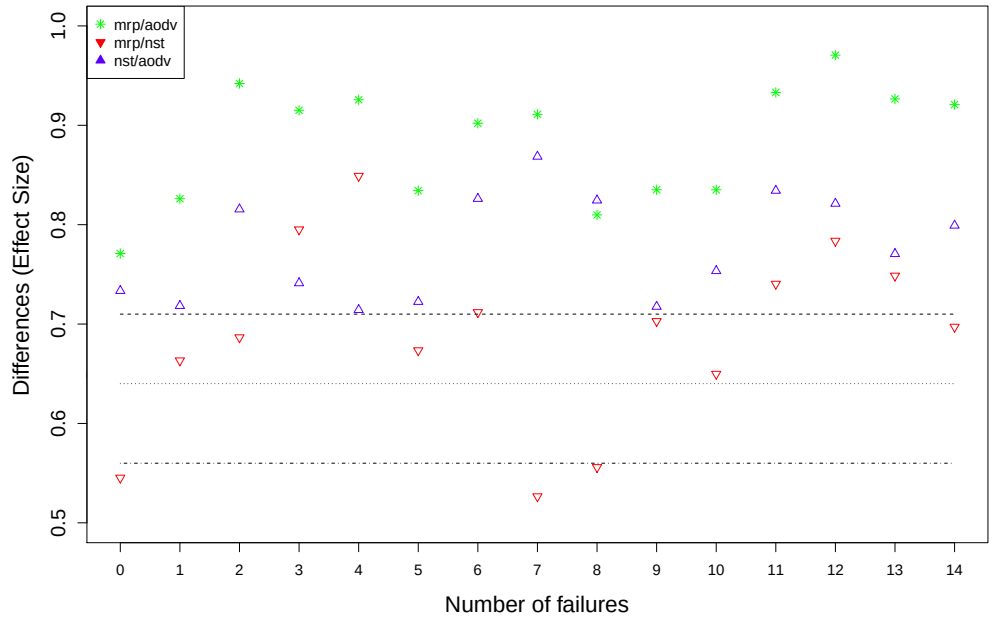



(a) The $A$-values for 0.5s

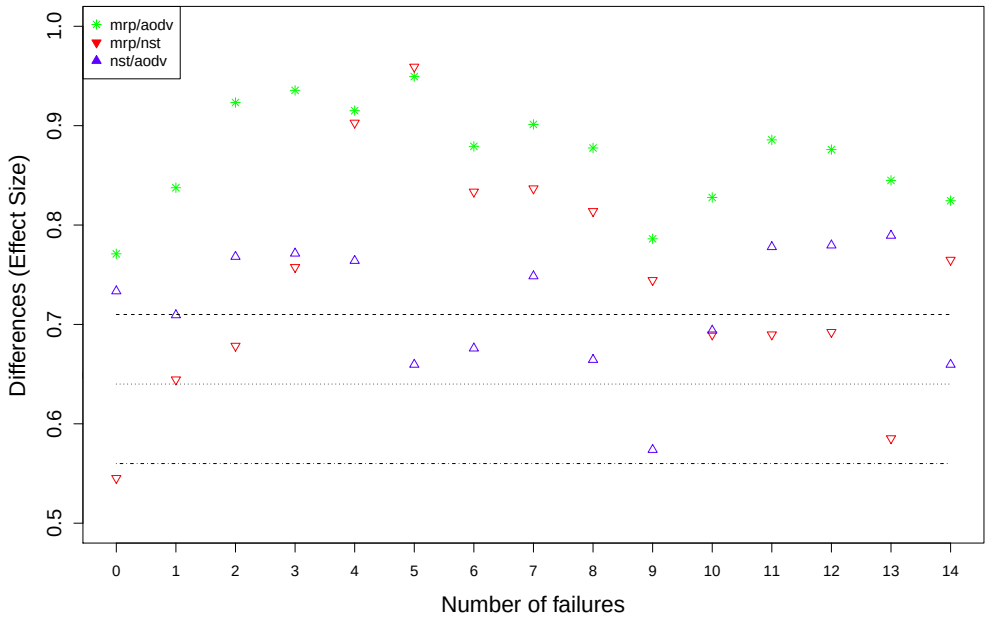



(b) The $A$-values for 2s

Figure 4.12: The $A$-values showing the scientific significant of energy difference.




(a) The $A$-values for 10s




(b) The $A$-values for 20s

Figure 4.13: The $A$-values showing the scientific significant of energy difference.

in Table 4.6. The higher energy consumption observed in AODV is due to the additional route discovery packets generated during packet collision. When more nodes are injected with failure, the energy consumption for the three routing protocols increases with the number of failure nodes, except for MRP when the failure duration is 0.5s. At 0.5s in MRP, the energy consumption is consistent at 2.73% with $\pm 0.03$ variation as shown in Table 4.6. AODV consumes the highest amount of energy consumption (from 3.95% to 12.5%) and MRP utilises the least amount of energy with a maximum of 8.09% when the number of failures is increase from 1 to 14. The difference in energy consumption observed between AODV and MRP is mainly due to the additional route request triggered during route discovery as the result of additional nodes failing. The broadcast nature of AODV during RD has injected a large number of control packets (shown by the higher number of routing overhead in Table 4.7), increasing the energy incurred during communication.

The energy consumption between MRP and NST also varies between the number of failures. The energy consumption for NST increases from 2.94% to a maximum of 9.30% when the maximum number of failures (14 nodes) are introduced as tabulated in Table 4.6. By managing between retransmission and route discovery, MRP has achieved a lower energy consumption than NST as the number of route discovery performed by MRP is less than NST. The *p*-values between NST and MRP from the statistical tests in Figure 4.10 and 4.11 show that their energy differences are statistically significant with the *A*-value above the medium effect size ($> 0.64$) in Figure 4.12 and 4.13. As for MRP and AODV, their differences have displayed a large effect size as all the *A*-values $> 0.71$. Hence, MRP is more energy efficient than AODV and NST.

**Routing Overhead**

Table 4.7 tabulates the median of the normalised routing overhead for different number of failures injected with difference failure durations. When no error is injected, AODV generates more routing packets than NST and MRP to recover route failure caused by collision. Further analysis on the simulation logs have revealed that the minimum routing packet required to send one packet to achieve the median PDR stipulated in Table 4.5 is 98 routing packets for 92.05% PDR in AODV, 52 routing packets for 97.16% PDR in NST, and 49 routing packets for 98.01% PDR in MRP. The number of routing packets on the network observed in AODV is doubled the amount of MRP. These routing packets have prevented the data packet to be transmitted successful resulting in a reduction in PDR. The

simulation logs have shown that the numbers of packet dropped due to collision (LQIs) are high in AODV.

When we introduced an error with 0.5s failure duration in to the nodes, the routing overhead of MRP is constant at 48 requests per data packet over the different number of failures as observed in Table 4.7 (RTR Load for 0.5s column). Although both MRP and NST rely on the retransmission mechanism to reduce the number of routing packets, the routing overhead for NST is significantly higher than MRP at 0.5s ($p$-value $< 0.01$ and $A$-value $< 0.71$). As more nodes are failed, the rate of routing packets generated by AODV (98 to 575) increase faster than NST (52 to 213). The routing overhead generated by MRP only increases (from 49 to 158 with 2s failure duration) when a failure longer than 2s is introduced.

To compare the significance between the different routing overheads, the $p$- and $A$- values are computed and presented in Figure 4.14, 4.15, 4.16 and 4.17. In Figure 4.14 and 4.15, the lower number of routing overhead generated by MRP are significantly difference to AODV and NST, with majority of the $p$-values are below the 0.05 significant line. Further statistical analysis reveals that the differences are also scientifically significant between MRP and AODV as the $A$-values in Figure 4.16 and 4.17 are above the large effect size (0.71) for all failure scenarios. Despite the small $p$-values between NST and AODV of less than 0.05 at 20s, some of the $A$-values are below 0.71 showing that not all of the difference are significant. On contrary, the differences between MRP and AODV are all significant. The $A$-values between MRP and NST also show a high level of effect size during high number of failures. Hence, we can conclude that the MRP requires less routing packets to deliver a data packet.

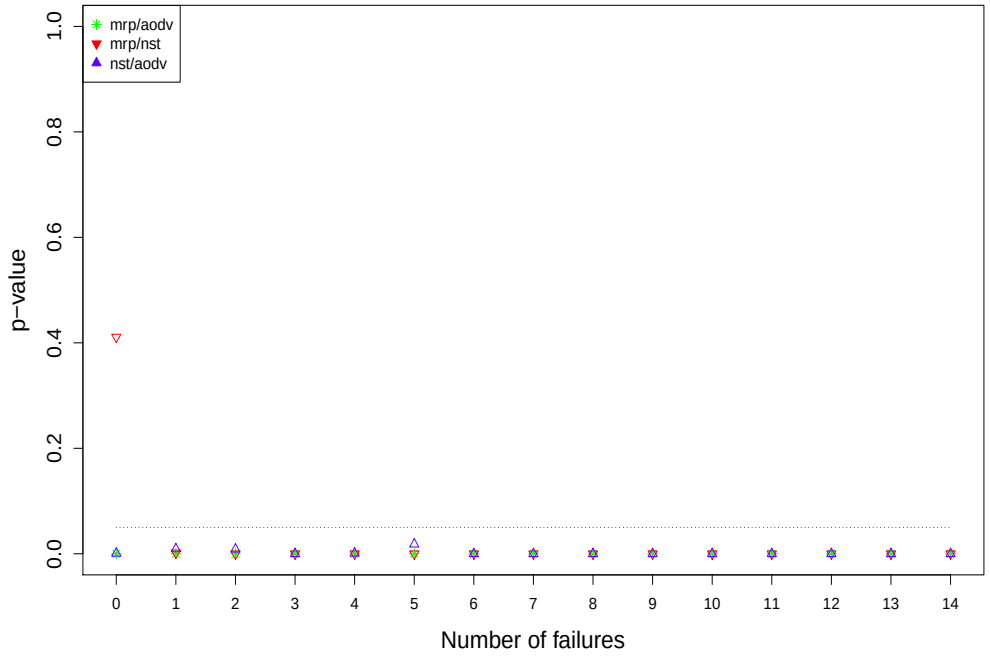
**Average Network Delay**

In Table 4.8, we tabulate the median of the average end to end packet delay for each of the routing protocols. The results show that the time required to deliver the packet increases with the number of failures. During error free condition, the delay between AODV, NST and AODV are not significant different with the $p$-values are greater than 0.05 in Figure 4.18 and 4.18.

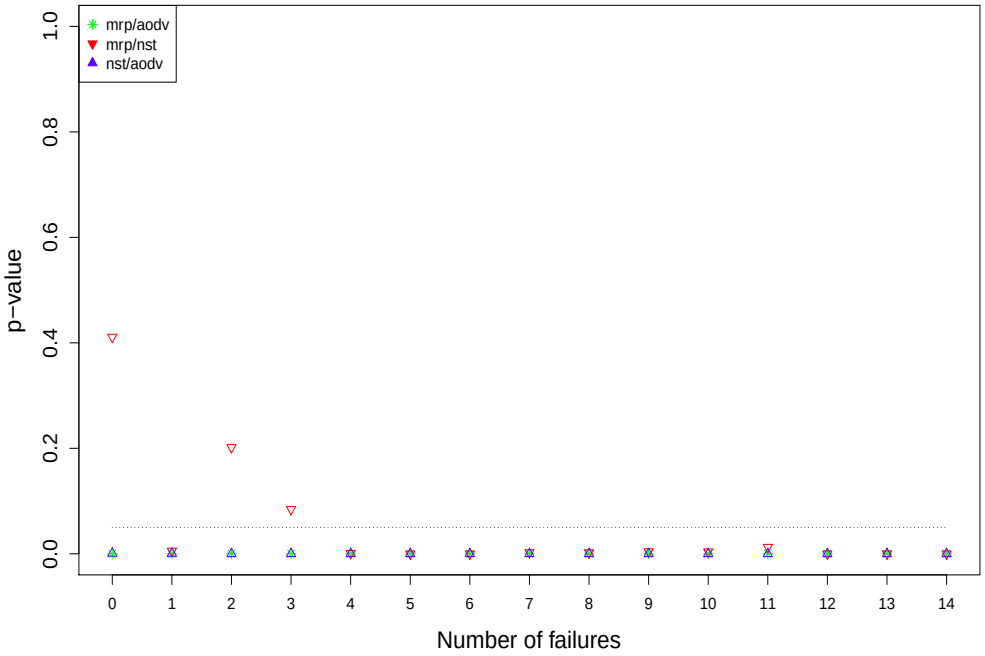
When 0.5s failure duration is injected to a node, the average delay for MRP remain unchanged. As more nodes are injected with 0.5s failure duration, the average delays for AODV and NST increase to 0.634s (from 0.106s) and 0.404s (from 0.083s) respectively. As the occurrence of failure becomes more frequent, the average time required to deliver a packet increases significantly for NST-AODV compared to AODV while the delay of MRP fluctuates between 0.087s and 0.127s. The delay in NST is double in AODV when more than seven nodes fail as tabulated

Table 4.7: The median of the routing packets required to deliver a packet calculated from 50 runs for different number of failures injected with different failure duration for MRP, NST and AODV.

| **Number of Failures** | **RTR Load for 0.5s** | | | **RTR Load for 2s** | | |
|---|---|---|---|---|---|---|
| | AODV | NST | MRP | AODV | NST | MRP |
| 0 | 98 | 52 | 49 | 98 | 52 | 49 |
| 1 | 138 | 92 | 49 | 146 | 82 | 50 |
| 2 | 149 | 92 | 48 | 248 | 81 | 73 |
| 3 | 284 | 106 | 48 | 224 | 94 | 64 |
| 4 | 206 | 126 | 48 | 290 | 138 | 79 |
| 5 | 186 | 143 | 48 | 283 | 142 | 73 |
| 6 | 246 | 129 | 49 | 328 | 144 | 82 |
| 7 | 262 | 150 | 49 | 276 | 171 | 128 |
| 8 | 365 | 132 | 48 | 341 | 171 | 130 |
| 9 | 425 | 136 | 48 | 366 | 185 | 138 |
| 10 | 446 | 126 | 48 | 336 | 138 | 79 |
| 11 | 456 | 186 | 48 | 358 | 199 | 154 |
| 12 | 495 | 206 | 48 | 472 | 244 | 143 |
| 13 | 507 | 195 | 48 | 465 | 248 | 154 |
| 14 | 575 | 213 | 48 | 474 | 245 | 158 |
| **Number of Failures** | **RTR Load for 10s** | | | **RTR Load for 20s** | | |
| 1 | 155 | 81 | 58 | 138 | 80 | 59 |
| 2 | 202 | 104 | 64 | 220 | 109 | 73 |
| 3 | 219 | 138 | 73 | 226 | 124 | 73 |
| 4 | 211 | 141 | 76 | 326 | 230 | 117 |
| 5 | 292 | 177 | 113 | 338 | 275 | 115 |
| 6 | 341 | 185 | 113 | 306 | 268 | 139 |
| 7 | 348 | 143 | 143 | 366 | 274 | 151 |
| 8 | 353 | 174 | 180 | 349 | 283 | 171 |
| 9 | 355 | 226 | 193 | 375 | 385 | 230 |
| 10 | 376 | 141 | 76 | 397 | 230 | 117 |
| 11 | 537 | 262 | 213 | 527 | 359 | 269 |
| 12 | 555 | 313 | 209 | 584 | 379 | 300 |
| 13 | 483 | 346 | 232 | 636 | 507 | 301 |
| 14 | 551 | 324 | 233 | 598 | 383 | 293 |

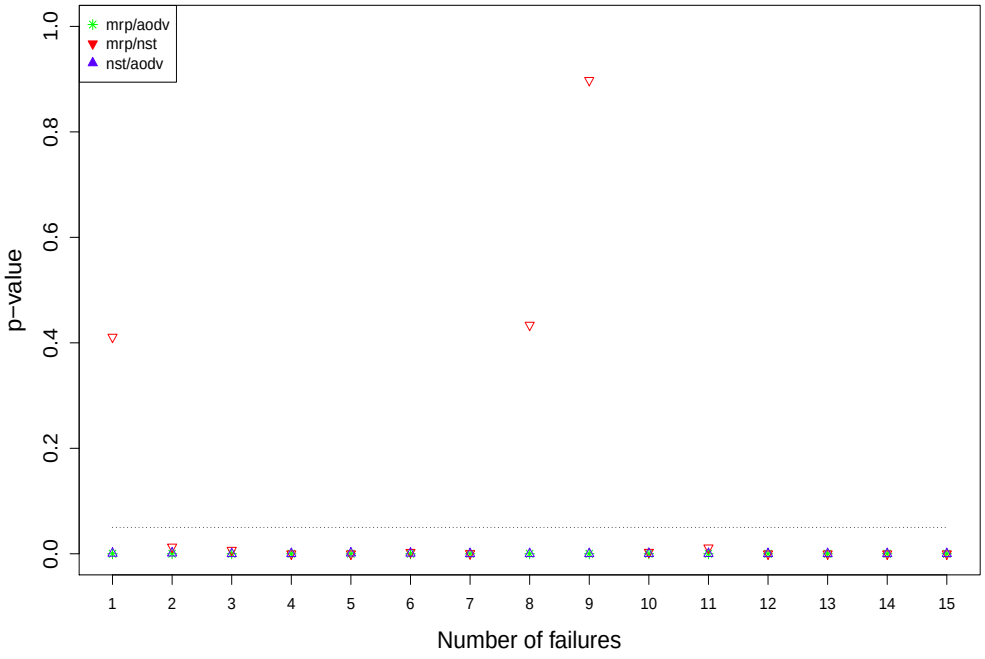



(a) The p-values for 0.5s

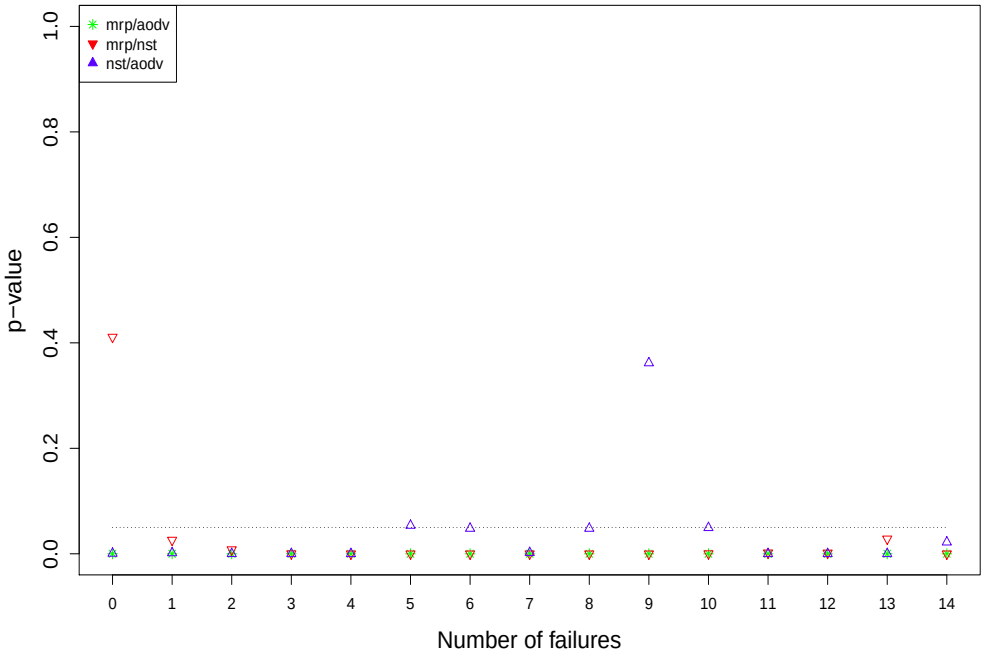



(b) The p-values for 2s

Figure 4.14: The *p*-values showing the statistical significant of routing overhead.

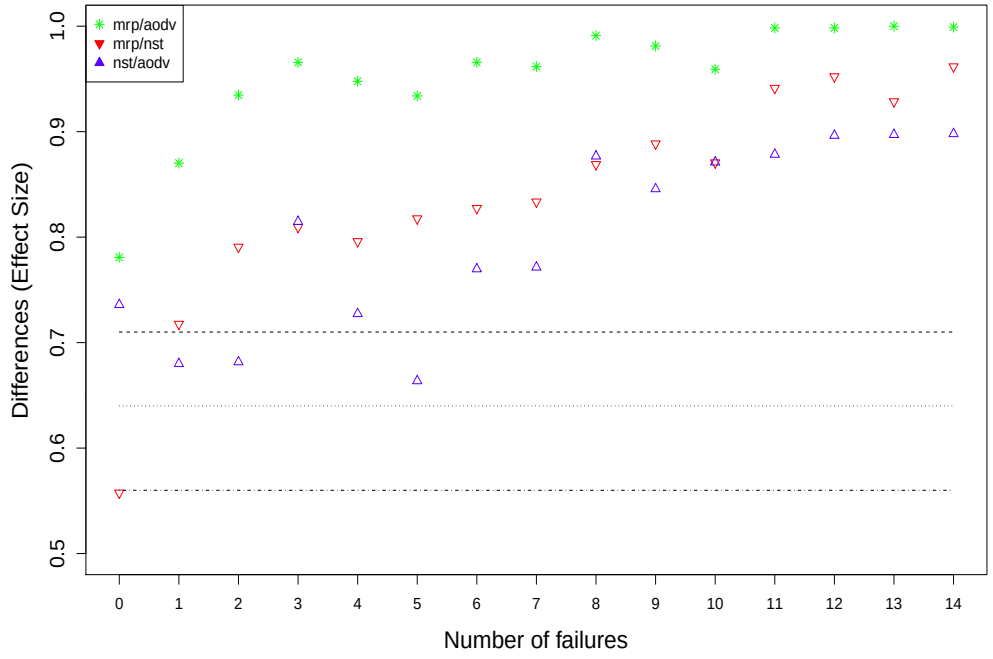



(a) The p-values for 10s

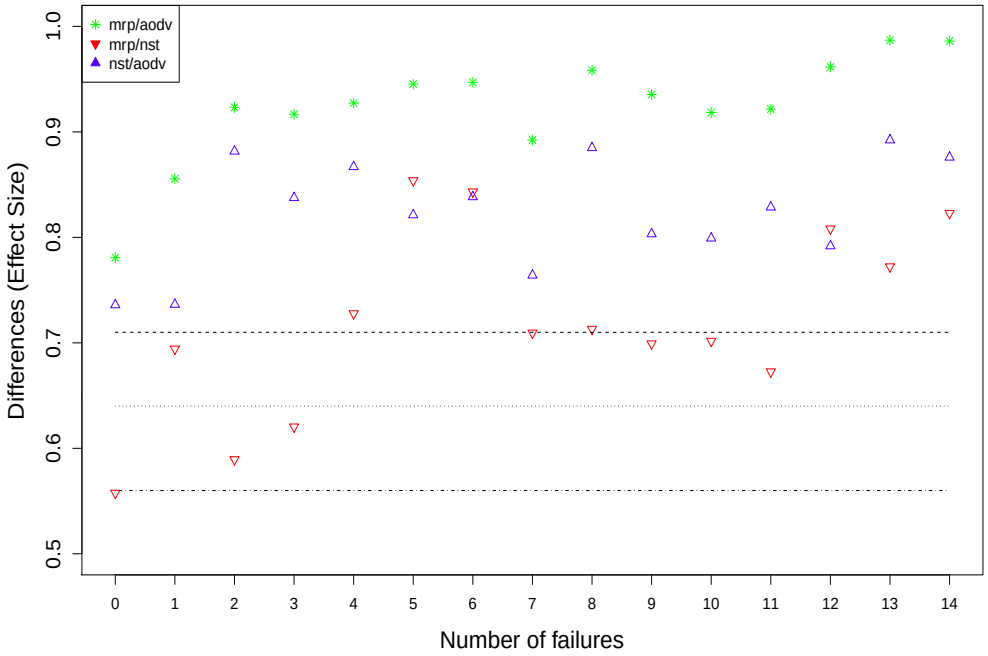



(b) The p-values for 20s

Figure 4.15: The *p*-values showing the statistical significant of routing overhead.

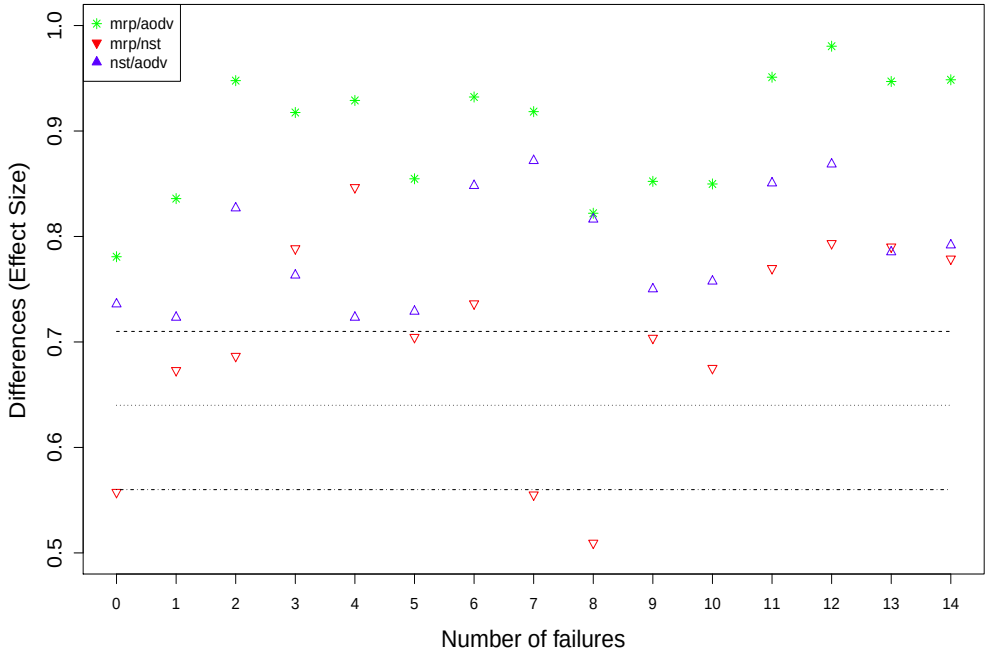



(a) The $A$-values for 0.5s

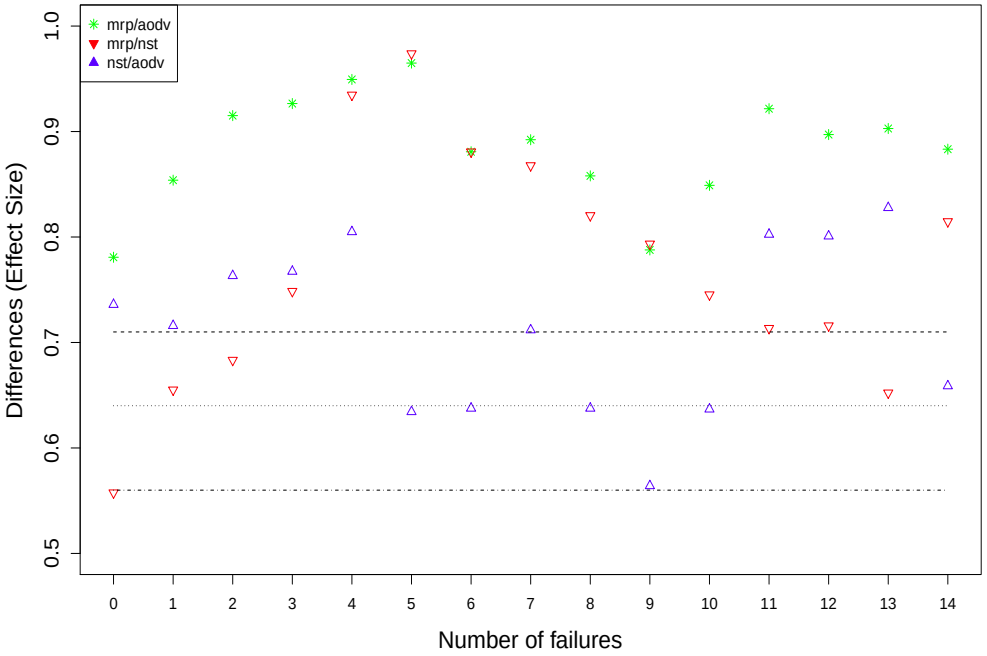



(b) The $A$-values for 2s

Figure 4.16: The $A$-values showing the scientific significant.




(a) The $A$-values for 10s




(b) The $A$-values for 20s

Figure 4.17: The $A$-values showing the scientific significant.

Table 4.8: The median of the average end to end delay for each packet calculated from 50 runs for different number of failures injected with different failure duration for MRP, NST and AODV.

| **Number of Failures** | **DLY (s)for 0.5s** | | | **DLY (s) for 2s** | | |
|---|---|---|---|---|---|---|
| | AODV | NST | MRP | AODV | NST | MRP |
| 0 | 0.106 | 0.083 | 0.087 | 0.106 | 0.083 | 0.087 |
| 1 | 0.116 | 0.118 | 0.087 | 0.138 | 0.097 | 0.105 |
| 2 | 0.128 | 0.130 | 0.105 | 0.280 | 0.102 | 0.140 |
| 3 | 0.204 | 0.128 | 0.087 | 0.256 | 0.119 | 0.149 |
| 4 | 0.257 | 0.165 | 0.112 | 0.332 | 0.206 | 0.154 |
| 5 | 0.255 | 0.230 | 0.105 | 0.374 | 0.234 | 0.143 |
| 6 | 0.346 | 0.220 | 0.112 | 0.333 | 0.213 | 0.149 |
| 7 | 0.446 | 0.214 | 0.087 | 0.324 | 0.213 | 0.233 |
| 8 | 0.459 | 0.154 | 0.112 | 0.333 | 0.256 | 0.328 |
| 9 | 0.484 | 0.331 | 0.127 | 0.323 | 0.404 | 0.342 |
| 10 | 0.500 | 0.165 | 0.112 | 0.452 | 0.206 | 0.154 |
| 11 | 0.632 | 0.309 | 0.107 | 0.397 | 0.420 | 0.389 |
| 12 | 0.619 | 0.307 | 0.087 | 0.468 | 0.429 | 0.449 |
| 13 | 0.551 | 0.281 | 0.087 | 0.467 | 0.563 | 0.401 |
| 14 | 0.634 | 0.404 | 0.079 | 0.572 | 0.387 | 0.403 |
| **Number of Failures** | **DLY (s)for 10s** | | | **DLY (s) for 20s** | | |
| 0 | 0.106 | 0.083 | 0.087 | 0.106 | 0.083 | 0.087 |
| 1 | 0.105 | 0.110 | 0.107 | 0.138 | 0.109 | 0.118 |
| 2 | 0.278 | 0.266 | 0.201 | 0.165 | 0.240 | 0.149 |
| 3 | 0.218 | 0.343 | 0.229 | 0.184 | 0.241 | 0.149 |
| 4 | 0.171 | 0.396 | 0.149 | 0.329 | 0.532 | 0.420 |
| 5 | 0.263 | 0.534 | 0.374 | 0.388 | 0.634 | 0.324 |
| 6 | 0.336 | 0.432 | 0.422 | 0.373 | 0.574 | 0.755 |
| 7 | 0.277 | 0.432 | 0.605 | 0.485 | 0.655 | 0.704 |
| 8 | 0.286 | 0.465 | 0.653 | 0.286 | 0.539 | 0.761 |
| 9 | 0.294 | 0.616 | 0.514 | 0.488 | 0.874 | 1.084 |
| 10 | 0.330 | 0.396 | 0.149 | 0.442 | 0.532 | 0.420 |
| 11 | 0.569 | 0.702 | 0.635 | 0.564 | 0.962 | 1.316 |
| 12 | 0.593 | 0.838 | 0.604 | 0.557 | 0.972 | 1.228 |
| 13 | 0.552 | 0.850 | 0.662 | 0.598 | 1.008 | 1.121 |
| 14 | 0.753 | 0.906 | 0.637 | 0.583 | 0.834 | 1.232 |

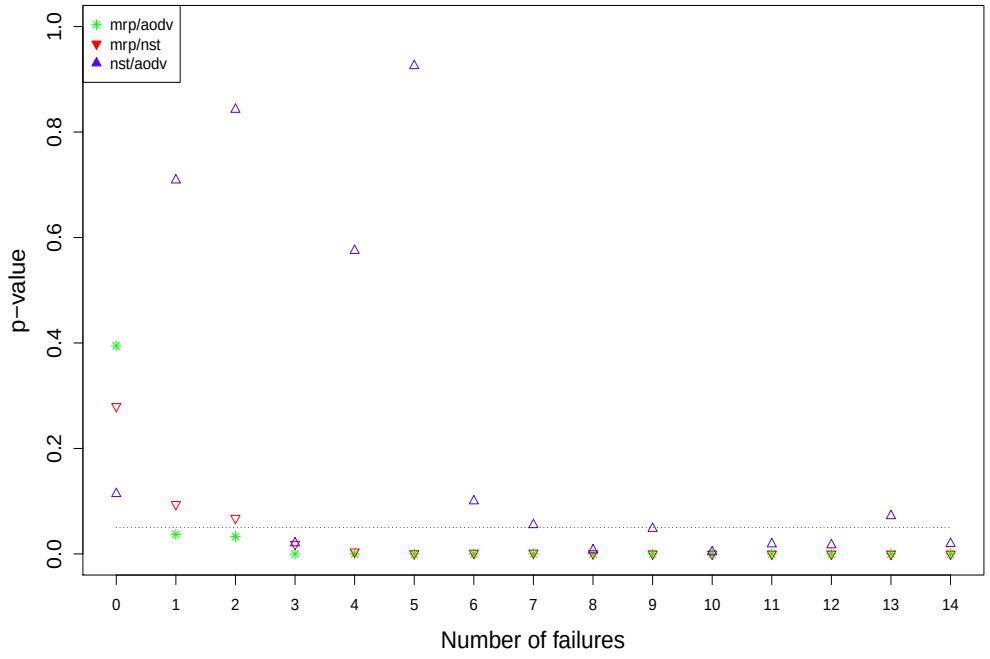
in Table 4.8. Statistical tests shown in Figure 4.18(a) reveal that the differences between NST and AODV are statistically significant when there are more than five nodes infested with failures. The $A$-values in Table 4.20(a) also show that the differences between NST and AODV have a medium effect size. The delay in MRP is also significantly lower than AODV and NST.

As the failure duration is increased to 2s and 10s, the time taken to deliver a packet increases with the number of failures for both NST and MRP in Table 4.8. Both MRP and NST require to buffer the packet while trying to retransmit the packet during failure. Hence, MRP and NST should have a higher delay compare to AODV. However, the $A$-values in Figure 4.20(b) for 2s and 4.21(a) for 10s show that the delay is similar for all the three routing protocols. Although the delay increases with the number of failures, the $p$-value $> 0.05$ and $A$-value $< 0.71$ show that the delays observed are not statistically different.

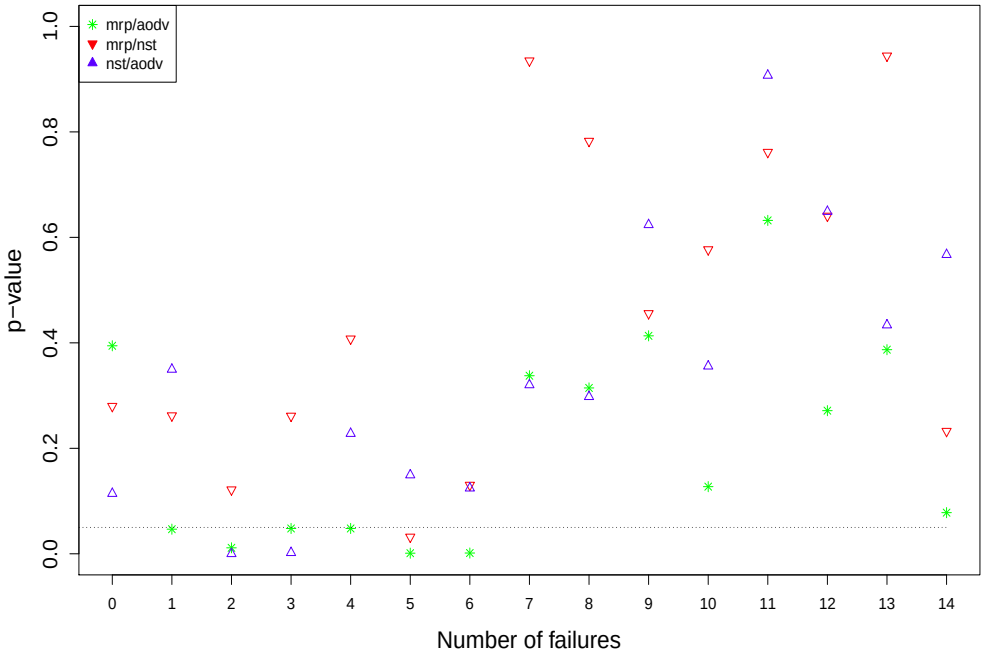
When the failure duration is further increased to 20s, the delay in MRP (0.087s to 1.232s) increases steeply compared to AODV (0.106s to 0.583s). However, the delay begins to saturate when more than nine nodes are injected. Both MRP and NST have to discover a new route each time retransmission fails when more errors occur. The delay in AODV is not always significantly lower than NST or MRP ($p$-value $> 0.05$, $A$-value $< 0.71$ when the number of failure is less than 4) although the data packet is sent immediately once a new route is established. Hence, MRP does not significantly increase the packet delay when compared to AODV and NST.

### 4.4.2 Discussion on the effect of failure size

From the performance results shown, we conclude that the performance of MRP is not affected by the number of node failures during short transient failures. When the failure duration increases, the PDR for the routing protocol decreases. However the use of multi-mode switching has significantly improved the performance of MRP compared to when applying AODV and NST alone. The combination of AODV and NST allows MRP to switch between retransmission when short transient error occurs and local route discovery during longer retransmission, or global discovery without compromising the network performance. In addition, AODV is less likely to be affected by the failure durations as any failure will lead to route discovery. Nevertheless, the performance of AODV degrades with the number of failures. The statistical tests have verified that differences in the PDR, energy consumption and routing overhead are both statistical and scientifically significant in all failure scenarios evaluated. Hence the MRP is more robust to different failure durations and failure numbers compare to the AODV and NST

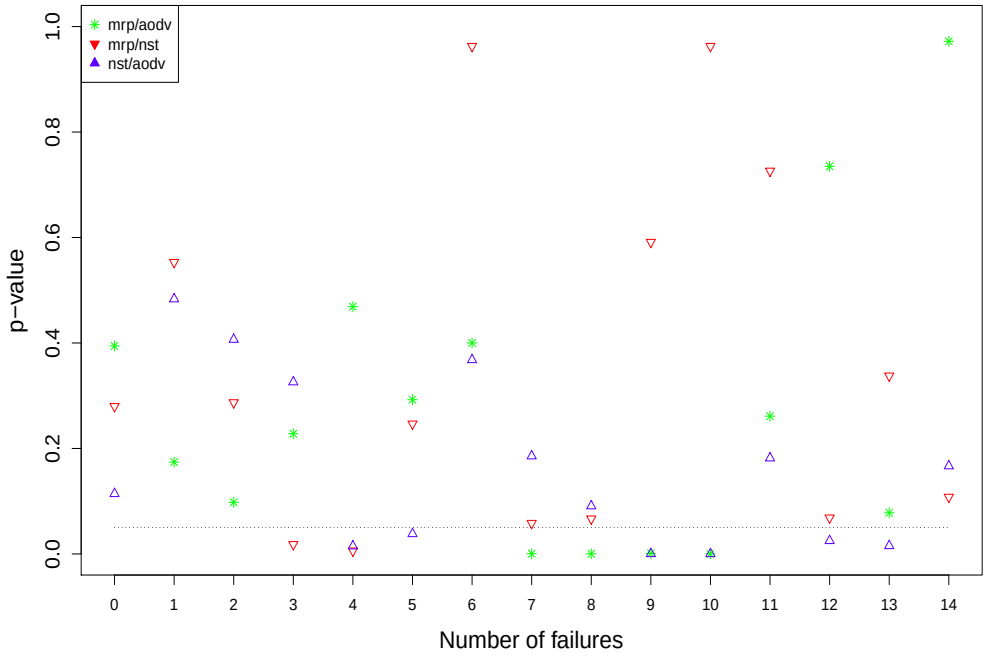



(a) The $p$-values for 0.5s

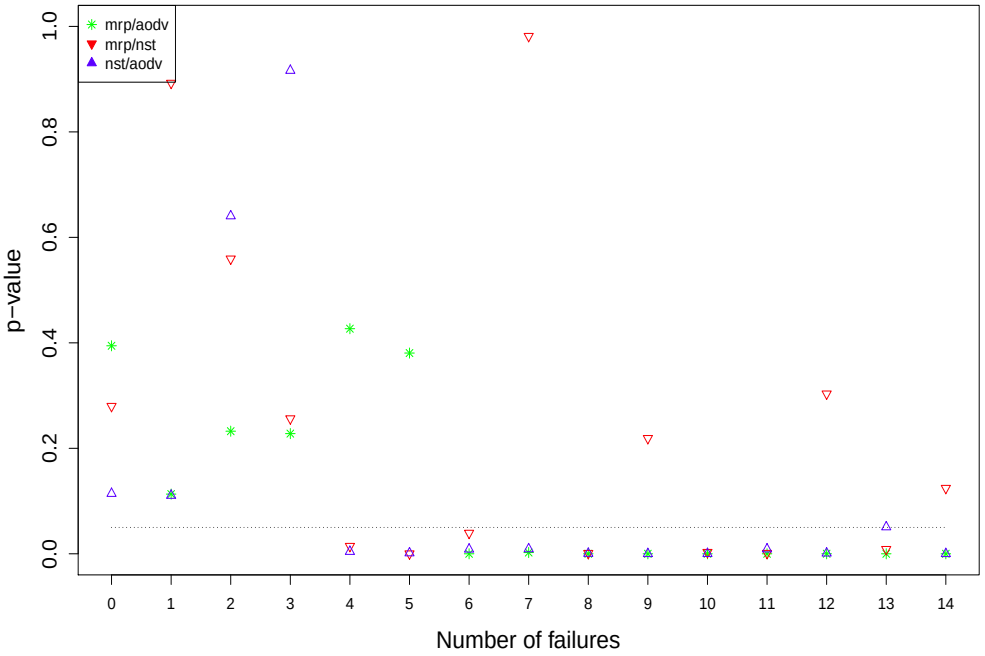



(b) The $p$-values for 2s

Figure 4.18: The $p$-values showing the statistical significant of average delay.

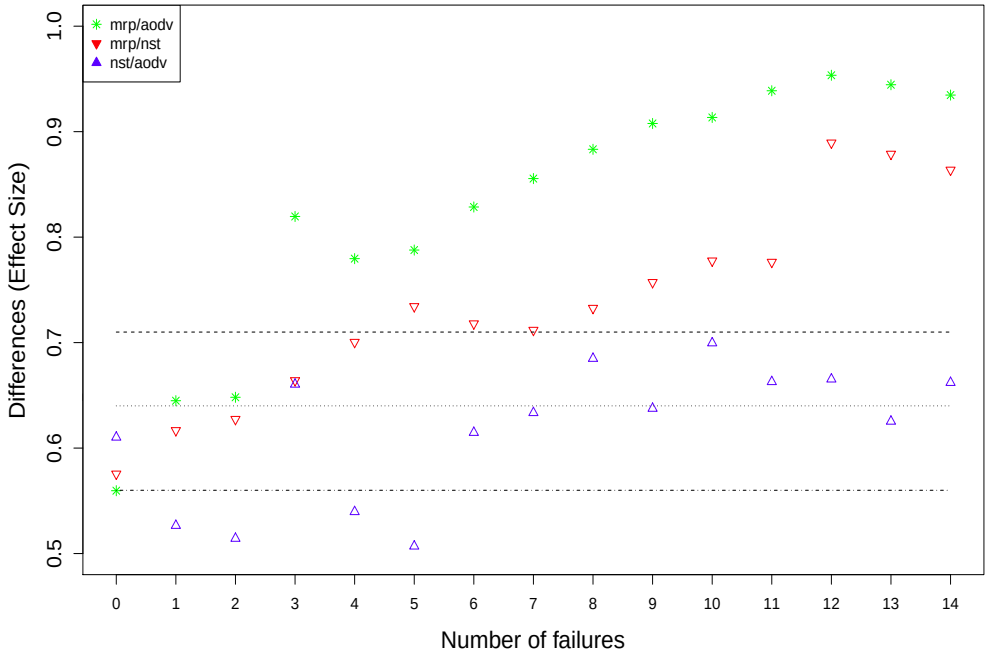



(a) The $p$-values for 10s

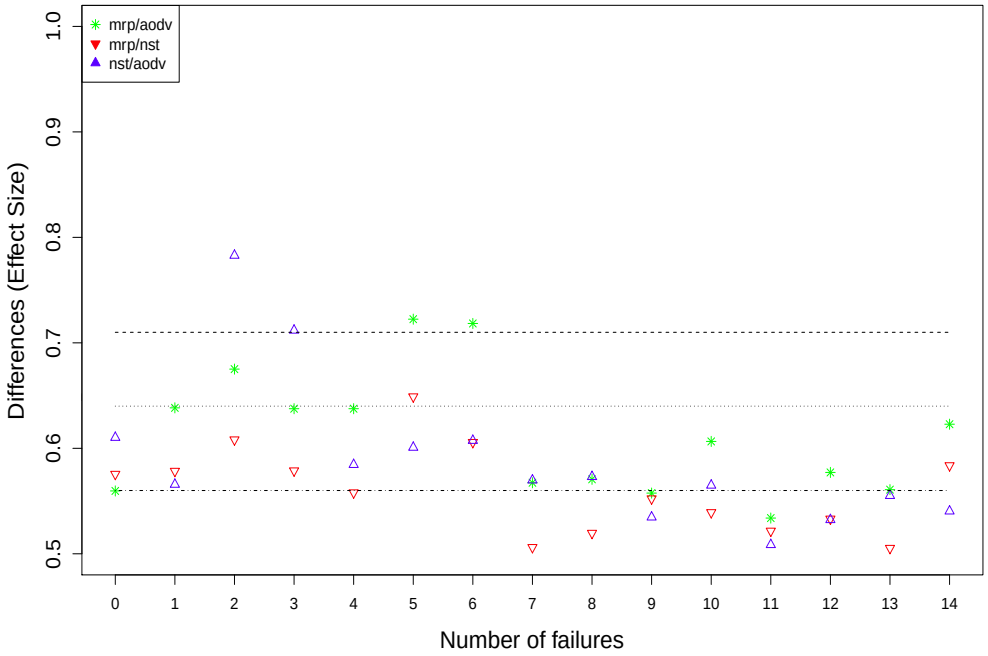



(b) The $p$-values for 20s

Figure 4.19: The $p$-values showing the statistical significant of average delay.

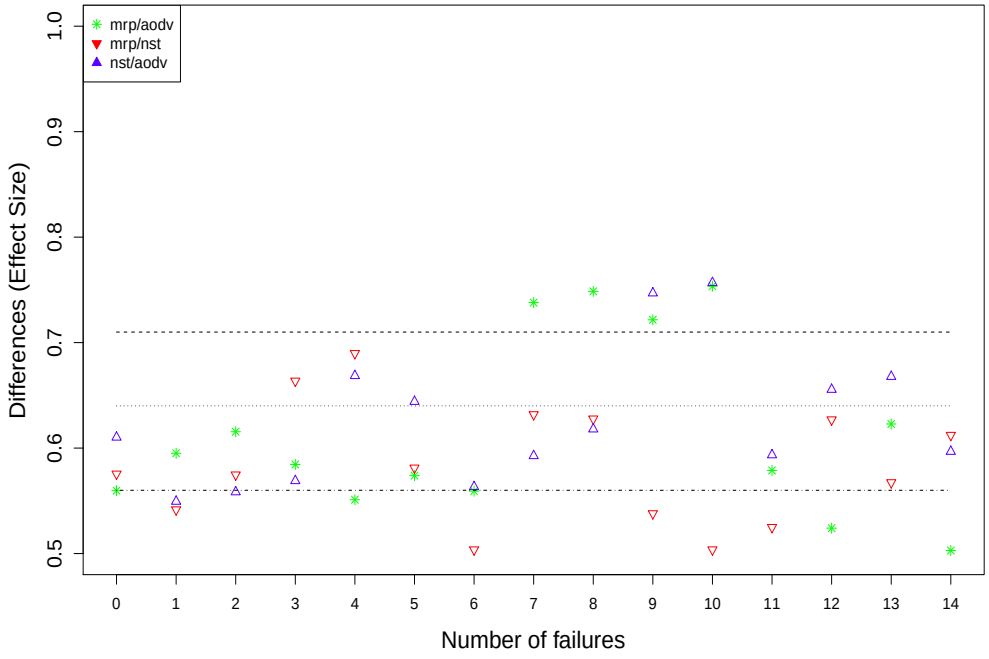



(a) The $A$-values for 0.5s

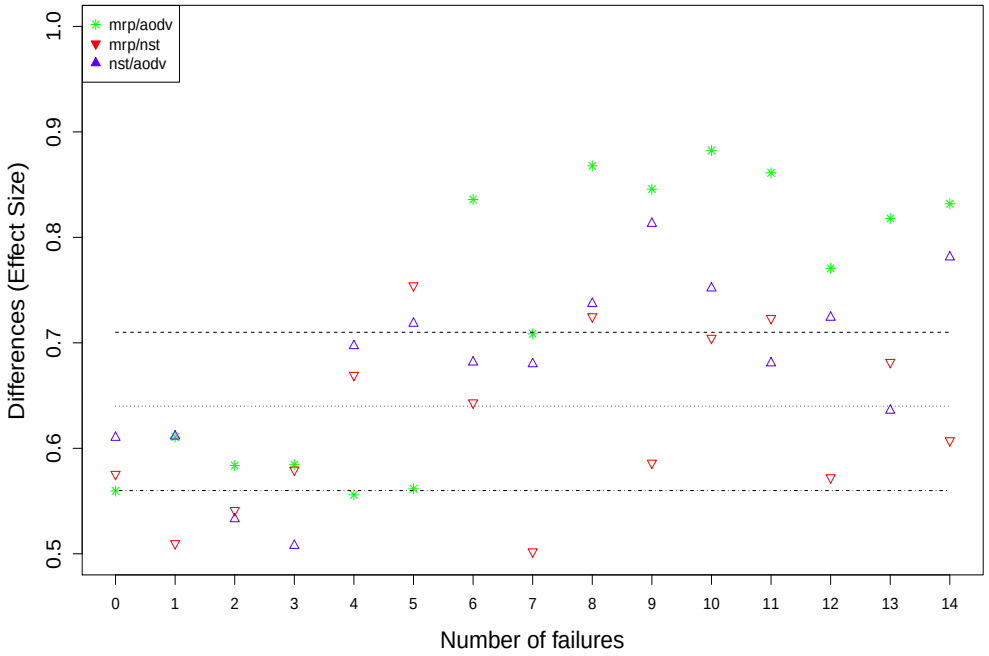



(b) The $A$-values for 2s

Figure 4.20: The $A$-values showing the scientific significant of average delay.




(a) The $A$-values for 10s




(b) The $A$-values for 20s

Figure 4.21: The $A$-values showing the scientific significant of average delay.

and is more efficient than NST and AODV. As the size of WSNs can scale up to a thousand nodes, we evaluate the scalability of the MRP in the next section.

## 4.5 On Scalability: The effects of network size

In this second set of experiments, the scalability of MRP is evaluated based on its ability to maintain its performance against AODV and NST. To evaluate the effects of different network sizes on the routing protocols, a predefined grid topology measuring X by X, where X = 5, 7, 10, 15, 400, 900, is used based on an outdoor application as shown in Figure 4.22. A grid topology is usually deployed outdoor in plantation as it is scalable up to 1000 nodes (Togami et al., 2012). This large network requires a multi-hop network protocol to route the packet from one end of the plantation to a base station. Hence it is appropriate to test MRP on this topology. The ON World survey on industrial WSNs reported by Hatler (2012) states that out of 74 applications, more than 50% of deployments have used less than 100 nodes while fifteen percent are between 100 to 1000 nodes. Based on that survey, we simulate 25 nodes as the minimum network size to ensure that the packet has traversed at least 3 hops across the network. For each experiment, the network size is doubled until it reaches the maximum of 900 nodes. Although some WSNs deployment may reach up to 1000 nodes, we did not test the MRP on more than 1000 nodes as NS-2.34 does not support network size that is larger than 1000 nodes (Schoch et al., 2008). A custom-built simulator is usually proposed to test a network with more than 1000 nodes. However, it can be difficult to validate the accuracy of the simulator (Bergamini et al., 2010). Hence, this approach is not taken. Furthermore, the industrial survey has reported that less than 5% industrial WSNs have deployed more than 1000 nodes (Hatler, 2012). Hence, the network size of more than 1000 nodes is not evaluated.

### 4.5.1 Packet Generation

Using the same simulation parameters specified in Section 4.3.3, different numbers of CBR are generated from different sources. The packet generated from different sources will traverse to one common sink as shown in Figure 4.22. The number of sources is configured according to the network size as a larger network is usually used to sense multiple areas. Based on a preliminary baseline experiment to determine the number of sources and the best position to place these sources without causing excessive packet collisions and congesting the network, a summary of the source and destination pair and the minimum number of hops


Figure 4.22: Network topology showing a square grid of 7 by 7 based on the outdoor plantation scenario

for each grid size is given in Table 4.9. This configuration may vary according to the network topology.

Table 4.9: The summary of the traffic flow patterns applied for different network sizes. The network topology for each of the grid sizes is given in Appendix A

| Grid Size | Source | Dest | $Min_{Hop}$ | *f(N)* |
|---|---|---|---|---|
| 5 by 5 | 1,3,5,15 | 25 | 8 hops | 3 |
| 7 by 7 | 0,2,4,14,28 | 48 | 12 hops | 5 |
| 10 by 10 | 2,4,6,20,40,60 | 99 | 18 hops | 8 |
| 15 by 15 | 0,2,4,6,8,30,60,90,120 | 224 | 28 hops | 13 |
| 20 by 20 | 2,4,6,8,10,12,40,80,120,160,200,240 | 399 | 38 hops | 18 |
| 30 by 30 | 2,4,6,8,10,12,60,120,180,240,300,360 | 899 | 58 hops | 28 |

### 4.5.2 Fault injection

To simulate the failure, a node is failed when it received a packet that has traversed more than $H_t/2$ hops to allow for local re-routing. $H_t$ is the maximum number of hops from the source to the destination. The number of failures (N) injected is set to be proportional to the size of the network as shown in Table 4.9 in order to have some effects on the routing protocol. From the initial test simulations, *f(N)* can be set according to $(Min_{Hop})/2 - 1$.

### 4.5.3 Results

The result in Table 4.10 shows that the networks with different number of nodes (25-, 49-, 100-, 225-, 400-, and 800-nodes) yield different performance values for each of the evaluation metrics. To show more clearly how the performance of MRP responds to failure as the network size increases, the box-whiskers plots for each of the evaluations metrics are provided in Figure 4.23 and 4.23. Table 4.11 shows the outputs generated from the statistical test from three different sets of results to show its significance.

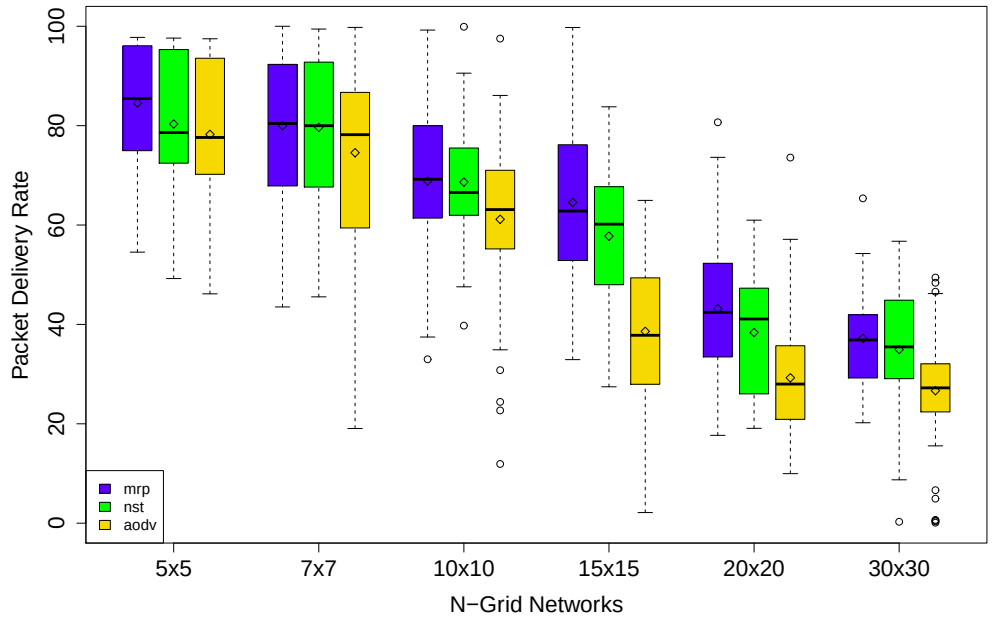
Figure 4.23(a) shows the percentage of the number of packets delivered successfully over the number of packets sent. If we analyse the PDR in Figure 4.23(a), we can see that the PDR of MRP is always higher than NST and AODV. For a small network, MRP delivers 85.45% packets compared to 78.60% in NST and 77.62% in AODV. As the network size doubles, the PDR rate decreases for all the routing protocols. When 225 nodes were deployed, the difference in the PDR for MRP to AODV is large where the difference in their median is 25% and their mean is 25.96%. According the Table 4.11, the difference in PDR for MRP and AODV in the network size more than 220 nodes are both statistical and scientifically significant ($p$-value=8.28e-11 and the $A$-value=0.88). The $A$-value always gives a large effect size ($A$-value $> 0.72$) when the network size is larger 225 nodes.

In term of the number of routing overhead, AODV generates more routing packets than MRP and NST-AODV. The number of route requests generated increases exponentially as the network size increases as shown in Figure 4.24(b). AODV generates more routing packets than in NST and MRP when the network size is greater than 225 nodes due to the propagation and regeneration of route requests by individual nodes. This high number of routing overheads depicted in Figure 4.24(b) reduces the PDR observed in Figure 4.23(a) as the nodes spend more time in transmitting the routing packets than the data packets. To send one packet, AODV uses three route request packets for a 25 nodes network. This route requests double (8) as the networks become twice in size. At 225 nodes, the route requests increase by a factor of four (from 16 requests to 69 requests) for AODV. This halves the PDR (from 63.11% to 37.81%).

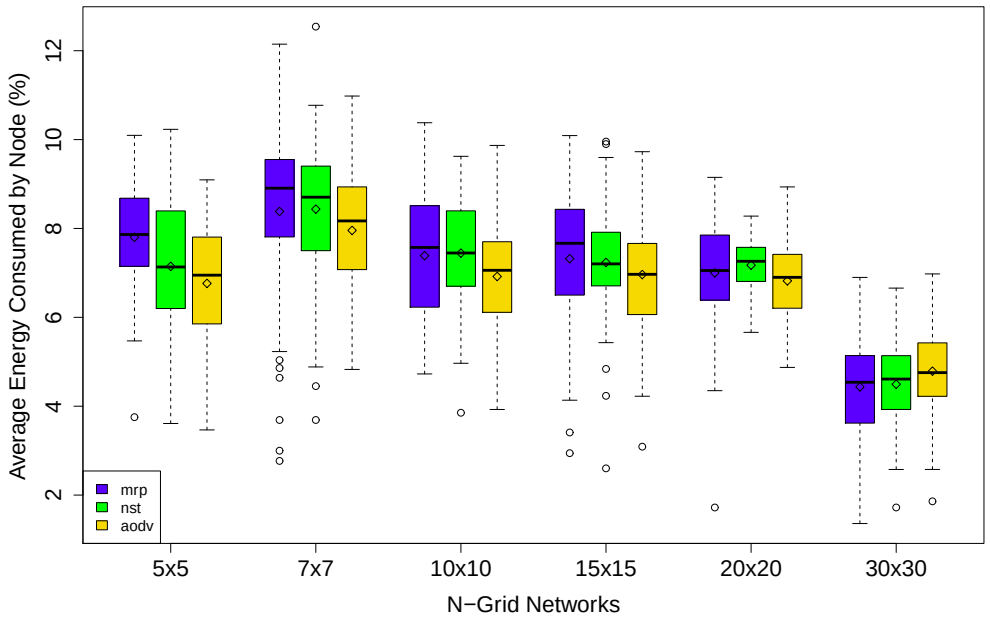
Figure 4.24(a) also shows a higher average end to end delay between AODV (0.93s) and MRP (0.51s). This difference is significant for a network of 225 nodes ($p$-value=0.01 with a medium effect size of $A$-value=0.33). This end to end delay is the implication of the additional route discovery initiated during node failure, where the packet is buffered. Due of this additional routing packet, normal packet transmission cannot be performed and the packet is dropped. Hence, less packet is being delivered as observed in Figure 4.23(a).

Table 4.10: The performance results in term of PDR, average energy remaining in a node, routing packet required to deliver a packet, and the average end to end delay for the three protocols: MRP, NST and AODV. Shown are the median and mean results for the network size of 25-, 49-, 100-, 225, 400, and 900-nodes exposed to *f(N)* failure nodes with 1 second failure duration.

| **Network Size(Nodes)** | | | **25** | **49** | **100** | **225** | **400** | **900** |
|---|---|---|---|---|---|---|---|---|
| *Median* | *PDR (%)* | *MRP* | 85.45 | 80.44 | 69.21 | 62.81 | 42.43 | 36.86 |
| | | *NST* | 78.60 | 79.99 | 66.54 | 60.18 | 41.09 | 35.48 |
| | | *AODV* | 77.62 | 78.18 | 63.11 | 37.81 | 28.00 | 27.23 |
| *Mean* | *PDR (%)* | *MRP* | 84.57 | 80.00 | 68.87 | 64.56 | 43.17 | 37.20 |
| | | NST | 80.33 | 79.66 | 68.64 | 57.76 | 38.37 | 34.95 |
| | | *AODV* | 78.24 | 74.55 | 61.15 | 38.60 | 29.25 | 26.68 |
| *Median* | *Average Energy Remain (J)* | *MRP* | 18.43 | 18.22 | 18.49 | 18.47 | 18.59 | 19.09 |
| | | *NST* | 18.57 | 18.26 | 18.51 | 18.56 | 18.55 | 19.08 |
| | | *AODV* | 18.61 | 18.37 | 18.59 | 18.61 | 18.62 | 19.05 |
| *Mean* | *Average Energy Remain (J)* | *MRP* | 18.44 | 18.32 | 18.52 | 18.54 | 18.60 | 19.11 |
| | | *NST* | 18.57 | 18.31 | 18.51 | 18.55 | 18.57 | 19.10 |
| | | *AODV* | 18.65 | 18.41 | 18.62 | 18.61 | 18.64 | 19.04 |
| *Median* | *Normalised Routing Load* | *MRP* | 2 | 3 | 8 | 17 | 41 | 195 |
| | | *NST* | 3 | 4 | 10 | 20 | 49 | 216 |
| | | *AODV* | 3 | 7 | 16 | 69 | 135 | 422 |
| *Mean* | *Normalised Routing Load* | *MRP* | 2 | 3 | 8 | 19 | 46 | 197 |
| | | *NST* | 3 | 4 | 10 | 23 | 58 | 1442 |
| | | *AODV* | 3 | 8 | 21 | 104 | 146 | 4434 |
| *Median* | *Average Delay* (s) | *MRP* | 0.04 | 0.29 | 0.57 | 0.51 | 0.85 | 0.39 |
| | | *NST* | 0.04 | 0.38 | 0.76 | 0.63 | 0.92 | 0.55 |
| | | *AODV* | 0.04 | 0.27 | 0.51 | 0.93 | 1.14 | 0.42 |
| *Mean* | *Average Delay* (s) | *MRP* | 0.14 | 0.49 | 0.78 | 0.64 | 0.95 | 0.59 |
| | | *NST* | 0.14 | 0.45 | 0.73 | 0.75 | 1.12 | 0.64 |
| | | *AODV* | 0.09 | 0.43 | 0.75 | 1.01 | 1.28 | 0.64 |

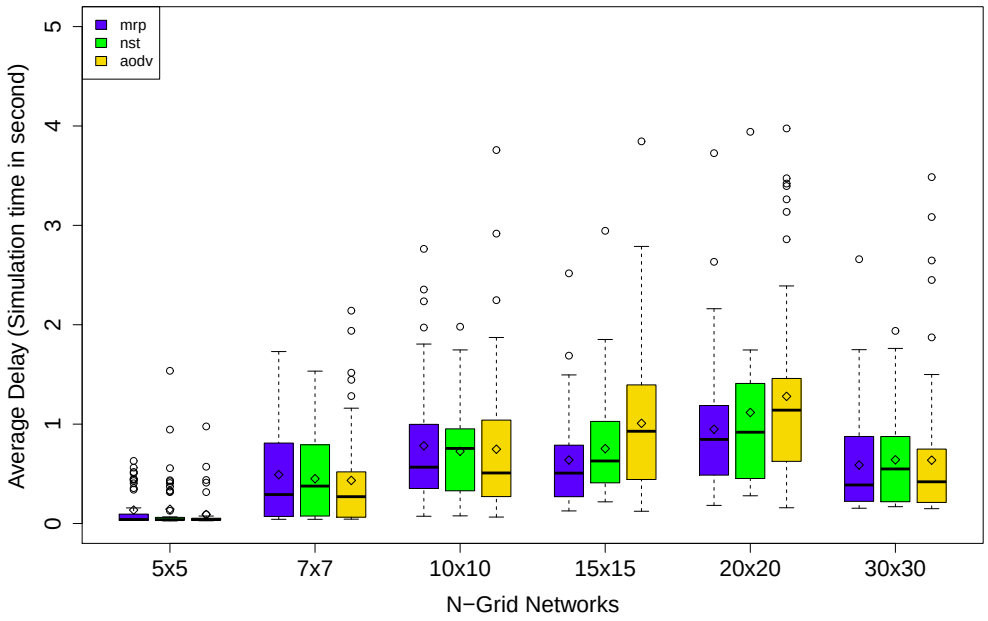



(a) PDR decreases with network size

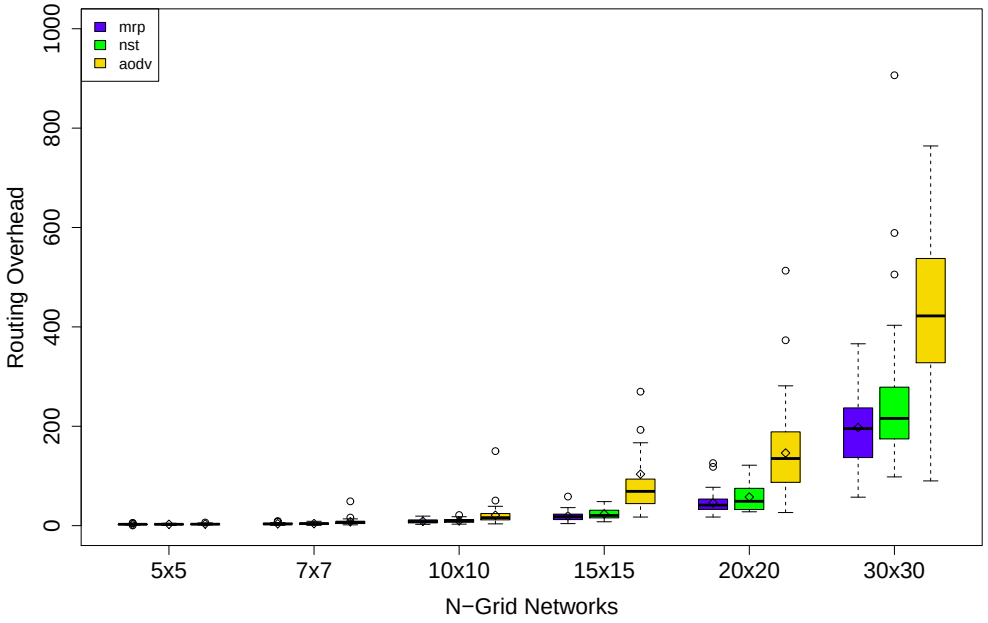



(b) Similar energy consumption is observed between MRP, AODV and NST.

Figure 4.23: Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for different network sizes with 17 failing nodes, with failure duration = 1s, failure interval = 5s.




(a) Similar average end to end delay observed for all networks.




(b) The routing packet generated increases exponentially as the network size increases.

Figure 4.24: Box-Whiskers plot showing the median, mean (diamond), lower quartile, upper quartile, highest and lowest values for different network sizes with 17 failing nodes, with failure duration = 1s, failure interval = 5s.

Table 4.11: *p*- and *A*-values indicating the difference between MRP and AODV, MRP and NST, and NST and AODV for different networks sizes. Numbers highlighted in ***bold-italic*** are significant, comprising a large effect

| **Significance Tests** | | **P-Test (*p*-value)** | | | **A-test (*A*-Value)** | | |
|---|---|---|---|---|---|---|---|
| **Metrics** | **No of (Nodes)** | **MRP AODV** | **MRP NST** | **NST AODV** | **MRP AODV** | **MRP NST** | **NST AODV** |
| *PDR* | *25* | ***0.02*** | 0.21 | 0.27 | 0.64 | 0.57 | 0.56 |
| | *49* | 0.14 | 0.87 | 0.21 | 0.59 | 0.51 | 0.57 |
| | *100* | ***0.04*** | 0.63 | ***0.04*** | 0.62 | 0.53 | 0.62 |
| | *225* | ***8.28e-11*** | ***0.05*** | ***1.43e-08*** | ***0.88*** | 0.61 | ***0.83*** |
| | *400* | ***0.01*** | 0.27 | ***1.66e-08*** | ***0.72*** | 0.41 | ***0.79*** |
| | *900* | ***1.12e-07*** | 0.63 | ***0.01*** | ***0.77*** | 0.53 | ***0.71*** |
| *Average Energy Remains* | *25* | ***0.01*** | ***0.03*** | 0.15 | ***0.29*** | 0.37 | 0.42 |
| | *49* | ***0.04*** | 0.70 | 0.05 | 0.38 | 0.48 | 0.39 |
| | *100* | 0.08 | 0.89 | 0.03 | 0.40 | 0.51 | 0.37 |
| | *225* | 0.06 | 0.29 | 0.23 | 0.39 | 0.44 | 0.43 |
| | *400* | 0.14 | 0.73 | 0.18 | 0.38 | 0.47 | 0.43 |
| | *900* | ***0.05*** | 0.80 | 0.13 | 0.60 | 0.51 | 0.58 |
| *Routing Load* | *25* | ***0.04*** | 0.43 | 0.12 | 0.38 | 0.45 | 0.41 |
| | *49* | ***7.29e-07*** | 0.20 | ***1.21e-05*** | ***0.21*** | 0.43 | ***0.25*** |
| | *100* | ***1.55e-09*** | 0.32 | ***8.84e-08*** | ***0.16*** | 0.44 | ***0.20*** |
| | *225* | ***8.03e-16*** | ***0.06*** | ***4.42e-14*** | ***0.03*** | 0.39 | ***0.06*** |
| | *400* | ***4.44e-06*** | 0.21 | **7.20e-18** | ***0.11*** | 0.60 | ***0.06*** |
| | *900* | ***1.27e-16*** | ***0.02*** | **4.70e-09** | ***0.09*** | 0.37 | ***0.18*** |
| *Average Delay* | *25* | 0.14 | 0.38 | 0.69 | 0.59 | 0.55 | 0.52 |
| | *49* | 0.60 | 0.97 | 0.42 | 0.53 | 0.50 | 0.55 |
| | *100* | 0.51 | 0.93 | 0.56 | 0.54 | 0.49 | 0.53 |
| | *225* | ***0.01*** | 0.10 | ***0.05*** | 0.33 | 0.41 | 0.39 |
| | *400* | 0.50 | 0.61 | ***0.03*** | 0.44 | 0.54 | 0.39 |
| | *900* | 0.61 | 0.69 | 0.40 | 0.53 | 0.48 | 0.55 |

### 4.5.4 Discussion on the scalability of MRP

In this section, we have reported that by increasing the number of the nodes in a network can reduce the packet delivery of the applications. However, the performance of MRP is significantly better than AODV in a large network. From Figure 4.23, the PDR of AODV decreases significantly compared to MRP when more than 255 nodes are deployed ($p$-value $< 0.05$ and $A$-value $> 0.83$ as shown in Table 4.11). The PDR of MRP is always higher than AODV and NST with a lower routing overhead. This significant packet drop in AODV is caused by RREQ packets generated by the node during failures. These packets multiply and propagate through the network, creating a sudden burst of traffic and causing other nodes have to wait for data packet transmission or drop the packet when the output buffer is full. The higher PDR and lower routing overhead in MRP have shown that the routing switching mechanism in MRP can increase the probability of the packet delivery by 5-10% avoiding the needs of unnecessary route discovery during failures. The result in Table 4.10 also shows that the delay and energy consumption between MRP, NST and AODV is the similar. The $p$- and $A$-values in Table 4.11 verify the means of the energy consumption and delay is not differences for MRP, NST, and AODV. Hence, MRP improves the reliability of packet delivery without incurring additional overhead for an outdoor WSNs application based on a grid topology.

## 4.6 Summary

In this section, we have performed an extensive set of simulations to test the performance and robustness of our proposed solution. Our experiments have demonstrated, through analysis and simulations, that a significant improvement in the number of successful packets delivered in the network can be achieved (Objective E4-1). The number of RDs are significantly lower than AODV and NST-AODV (Objective E4-3), making the networks less congested and more energy efficient (Objective E4-2). The results also show that switching delay introduced in the node are not statistically significant (Objective E4-4). Hence, the proposed multi-modal approach can achieve a better packet reliability and is more energy efficient than single mode approach. However, the MRP can only detect and recover from failure. It does not identify the failure that may be required to rectify the failure effectively and quickly. In the next chapter, we will investigate the ability of a bio-inspired algorithm to identify the faults and assist the multimodal recovery approach to respond according to the failure.

CHAPTER

# FIVE

# Assisted Recovery with An Immune-Inspired Classifier

This chapter investigates the application of Artificial Immune System (AIS) to assist in detecting and recovering from failures in WSNs. Most fault tolerance approaches applied in WSNs in Chapter 2 demand high resources, require offline training and centralised detection. Some researchers attempt to improve the network availability and reliability through redundancy (Zou and Chakrabarty, 2007), or by detecting the fault with limited automated recovery (Chandola et al., 2009). It is sometimes necessary to determine the cause of the fault online in order to rectify the problem quickly and effectively to reduce network downtime (Candea et al., 2004). As a result, researchers have applied the bio-inspired approaches such as the immune-inspired algorithms to solve complex engineering problems using one or two features of the immune system derived from an analogy of the application process and the biological principles (Hart and Timmis, 2008). The immune system has many attractive properties that are similar to WSNs such as self-adaptive, self-healing, robust, scalable, autonomous, and self-stabilisation that can be applied to resolve complex problem such as fault detection (Wallenta et al., 2010). The ability of the immune cells to self-detection, self-identification and self-recovery has provided an inspiration for the application of the immune-inspired algorithm to provide self-healing in WSN.

The objectives of this chapter are to

- Investigate and exploit how the self-healing property of the immune system can assist in identifying and recovering from radio anomalies.

- Apply an immune-inspired algorithm to provide automated diagnosis and recovery in WSNs to improve the reliability and efficiency of the routing protocol.

- Evaluate the performance of the automated detection and recovery system in both hardware (TelosB) and software (NS2).

We propose an immune-inspired Interference Detection and Recovery System (IDRS) that allows individual nodes to detect, diagnose and recover from network failure due to radio interferences. Radio interference can affect the communication in any wireless network that needs to be identified. Section 5.1 discusses why existing approaches are not suitable to distinguish interferences in WSNs and the self-adaptive and learning properties of an immune system are required. Based on the review of the different AIS algorithms in Section 2.7.3, the RDA has been selected to perform the interference classification due to its ability to detect anomalies in a dynamic changing environment (Hilder et al., 2012). The RDA is described in Section 5.2. The architecture and design of IDRS is presented in Section 5.3. The accuracy, efficiency and reliability of IDRS are evaluated and discussed in Section 5.4. Section 5.5 ends the chapter with the summary.

## 5.1 Motivation

Network failures such as having an unreliable link and packet dropped are common in WSNs as the node shares the same radio frequency with other radio emitting devices such as laptop, tablet and game console. The radio on a node is sensitive to the interference generated by these devices. It is necessary to scan the radio channel for any abnormal radio signal strength that is higher than its current radio signal and avoid transmission during interference. This function is usually provided by the link layer (Gutierrez et al., 2001). Before a node transmits a packet, CSMA/CA requires the node's radio to listen and detect for an idle channel. If the channel is clear, then the radio switches into the transmission mode and sends out a packet. This function only detects the presence of interference but does not determine the characteristics of the interference required for an effective recovery. The level of interference is dependent on both the *strength* and *duration* of Radio Frequency Interference (RFI) (Liu et al., 2010). It is easy to determine the *strength* of the interference (strong or weak) by measuring the radio signal strength indicator (RSSI) from the radio interface. As these interferences depend on the device usage pattern, it can be a challenge to determine

the interference's *duration* that can be short, long, transient or permanent and derive a pattern that can be used by the routing protocol to identify the interference characteristics and applied appropriate recovery or avoidance actions. It is necessary to collect interference patterns and classify the interference according to the strength and duration. From the classification, the appropriate response to rectify the fault generated by each interference type can be mapped onto each interference class and the detection system can use this classification information to identify and rectify the failure.

### 5.1.1 Problem Formulation

Lin et al. (2009) classify these interferences into three distinct patterns namely: small fluctuation created by multi-path fading of wireless signals; large disturbance due to shadowing effect of the presence of obstacles; continuous large fluctuations caused by WLAN devices. Each of these interference patterns can have different detrimental effects on the Packet Sending Ratio (PSR). Recent work by Wang and Akyildiz (2011) has shown that the interference from a WLAN network can produce up to 30% packet losses in WSNs. To guarantee the *availability* of the network, the node may need to detect and provide a fast and accurate response to overcome the interference. According to Candea et al. (2004), the *availability* can be formulated in terms of the relationship between Mean-Time-To-Fail (MTTF) and MTTR:

$$Availability = \frac{MTTF}{(MTTF + MTTR)} \times 100 \tag{5.1}$$

If the MTTR in Equation 5.1 minimises to zero, near 100% availability can be achieved. Hence, it is necessary to reduce the MTTR by improving the responses through a diagnostic process. Poor diagnosis may yield false positive and produces incorrect treatment that may consume unnecessary resources and increase recovery time.

Different recovery approaches can be taken by the routing protocol such as to retransmit, or varying of transmission power in order to communicate with the neighbour. In severe cases, the node may need to establish a new route in order to send the packet. Some of these recovery approaches such as flooding are more expensive to execute and time consuming than others. To illustrate, six static homogeneous nodes are deployed in the topology shown in Figure 5.1. Each node can forward packets and produce network statistics. The communication between a node (5) and its neighbouring nodes (4 and 7) are disrupted when a laptop placed close to the node (5) decides to transmit a packet randomly. Although the laptop employs the CSMA/CA to detect for an idle channel, it is unable to sense the

low power transmission of the WSN's node. This radio activity causes packet collision in the WSNs. As a result, the affected node will attempt to rectify the failure by executing protocol-specific recovery functions such as retransmission, flooding and collision avoidance (Perkins and Royer, 1999). Incorrect response can aggravate a congested network. For example, retransmission is best applied when the interference is temporary and transient. Retransmission can improve the probability of successful transmission and improve the reliability of the network (Balakrishnan et al., 1997). Local discovery should be avoided in a noisy network as it may lead to broadcast storm (Hsu et al., 2007). Hence, it is not only important to detect the presence of an anomaly, but also the cause of an anomaly needs to be established in order to make accurate and automated recovery decision and improve availability.

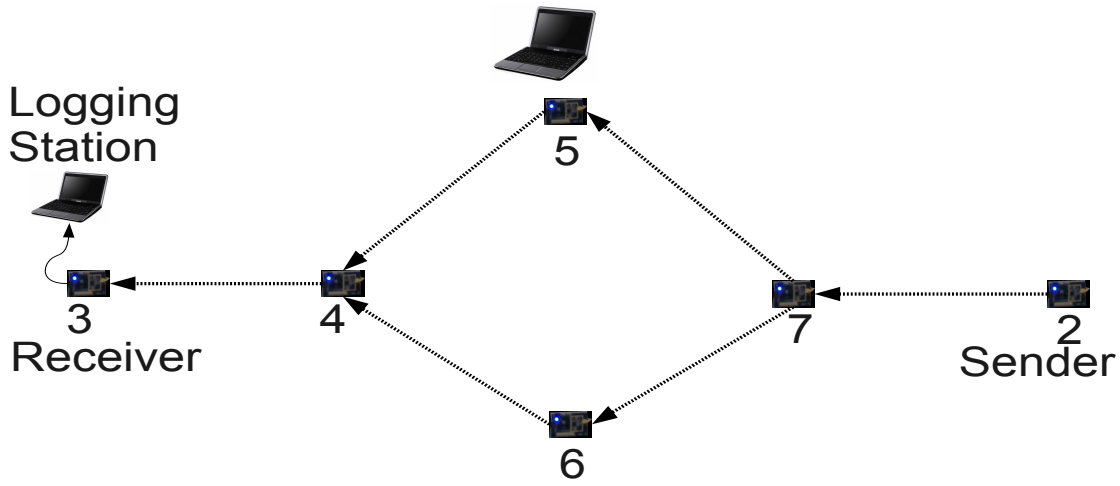



Figure 5.1: Interference source is introduced near node 5 to disrupt the radio communication between node 2 and 3.

### 5.1.2 Existing approaches

To perform the error processing techniques (error detection, diagnosis and recovery) described in Section 2.6 in a node can be difficult and expensive. Due to limited constraints in the node, the diagnostic and recovery mechanism taken should be performed online. It should also be low cost and low overhead with a high probability of rectifying the problem. Zacharias et al. (2012) propose the use of signal processing approaches such as the Fast Fourier Transform (FFT) to distinguish the radio interference with a known fixed frequency such as Microwave from Bluetooth and WLAN. However, the FFT is only performed in a computer using the RSSI collected from the sensor nodes via the base station as FFT is a

computationally expensive operation to be performed in the motes. For an $n$ elements series, the FFT algorithm computational complexity is $O(n^2)$ (Trinidad and Valle, 2009). Ong et al. (2010) show that the execution time can take from 20.39ms to 166.68ms in a MSP430 microcontroller depending on the number of operations performed and the input representation (an integer or floating point). With the stringent energy budget in sensor mote, a simple and quick algorithm over complex algorithms is more favourable to reduce execution time and energy consumption. Existing work mainly focus on anomalies generated by malicious attacks that usually have a static distinctive feature that can easily be classified (Ngai et al., 2006). Little work has investigated anomalies due to the presence of interferences in the operating environment such as WLAN communication. One of the challenges in detecting anomalies due to these interferences is that the duration and occurrence for these types of anomalies are unpredictable and varies with time (Liu et al., 2010). The duration and occurrence of these interferences are usually dependent on the type of radio devices or applications, and their usage pattern. Thus, it can be very difficult to be detected and classified using existing conventional statistical approaches (Boers et al., 2010).

### 5.1.3 Contributions

To diagnose a fault, the state of the faulty system must be collected and analysed when the fault occurs for immediate recovery. As only limited historical data can be stored in the memory for error processing, diagnostic information is usually sent to the sink for analysis. Due to unreliable communication, this information may be lost during delivery. There is a need to perform online fault diagnosis on the node. The importance of integrating the detection, diagnosis and recovery into one system has been widely ignored in literature as each fault tolerance approach is usually performed and analysed in isolation.

The contributions of this chapter are:

1. We propose an immune-inspired solution that allows individual nodes to self-detect, diagnose and recover from network failure. Cohen (2004) postulates that the immune system does not only protect the body. It also performs body maintenance through the process of recognition, cognition and response with respect to its current environment (Cohen, 2004).

2. We apply a multi layer approach where MRP provides the first line of defence to detect the fault and integrate the Receptor Density Algorithm (RDA) as a secondary defence to determine the interference in order to assist in recovery. The RDA is an AIS algorithm based on a T-Cell signalling model

with statistical kernel density function to detect anomalies (Owens et al., 2012).

3. We implement the RDA in real TelosB mote to recognise different types of interferences according to the duration and strength, affecting the PDR of the networks and perform as systematic evaluation to measure its accuracy.

## 5.2 The RDA: Receptor Density Algorithm

Based on the review in Section 2.7.3, we apply the RDA to perform the fault detection and identification in order to assist in recovery. RDA is a suitable for our problem as it has the ability to learn and discriminate between the normal and abnormal autonomously in a dynamic environment. It has also been shown to yield a high accuracy rate when apply to detect anomalies (Owens et al., 2012). It operates independently and can adapt to changes. This section provides an overview on the biological inspiration and the development of the algorithm.

### 5.2.1 Biological principle

In adaptive immunity, the T-Cell must perform a discrimination process to distinguish between the self and non-self antigens. The discrimination process is performed based on the population of the TCRs ability to bind to a population of antigens presented by the APCs. The discrimination does not depend solely on molecular recognition between the TCRs and antigens, but also on the stimulation received from all the antigens and the internal signalling of the T-Cell (Owens et al., 2012). The internal component of the TCR undergoes a process known as kinetic proofreading with a negative feedback. The binding between the TCR and antigen consists of energy consumption process that increases when a sufficiently strong and lasting bonding occurs. The antigens may continually associate and dissociate from the TCRs and the binding strength can be weak, medium and strong. Weak binding has little effect on the T-Cell, the medium binding inhibits the T-Cells and the strong binding will activate the T-Cells. These binding processes also involve negative energy consumption steps that are reversed as the binding dissociate. As the kinetic proofreading steps progress, a negative signal is released when the proofreading steps are greater than the base negative feedback. The antigen may also bind to the nearby receptors on the T-cell that inhibits and reverses the progress if a TCR dissociates from the antigen. The negative feedback generated by a TCR can also spread and dampen nearby TCRs (Owens et al., 2012). The spreading of the negative signal is a key concept in

RDA in determining the activation of the T-Cells. If no TCR can generate an activation signal, the negative feedback will pull the kinetic proofreading below the base negative state. If a small set of strong antigens associated to the TCR is able to complete the kinetic proofreading before the majority weaker antigens reach the negative feedback base state and reverse the kinetic proofreading, the T-Cell is activated, triggering immune response.

### 5.2.2 The Algorithm

Using the concepts introduced in the last section, Owens et al. (2012) extracts the computational abstraction of the TCRs and develop an algorithm with an abstracted molecular recognition system for detecting anomalies. The algorithm begins by defining the term *receptor* taken from Owens et al. (2012).
**Definition.** *A receptor r is a tuple (p, n, $\beta$, $\ell$, c), where:*

*$p \in [0, \ell]$, the receptor position;*
*$n \geq 0$, the generated negative feedback*
*$\beta > 0$, the base negative feedback barrier;*
*$\ell \in (0, \infty), \ell > \beta$, the length of the receptor;*
*c = {0,1}, the receptor output. c = 1 if $p \geq \ell$;*

This definition of a receptor is *an abstraction of the internal component of the TCR* in which $p$ represents the kinetic proofreading state of the internal component of the TCR and $\ell$ is the maximum kinetic proofreading state that is capable to generate an activation signal. $c$ is the output that determines whether the T-cell is activated. $n$ represents the generation of negative feedback in the neighbourhood of the receptor and the threshold $\beta$ is the base negative feedback barrier (Owens et al., 2012).

The behaviour of a receptor presented in Figure 5.2 is described as follows:

- A receptor receives a sequence of input $\{u_t\}$ with $u_t \in \mathbb{R}$ at discrete time $t = 0, 1, 2, ...$ and $u_t \geq 0$.
- The input $u_t$ pushes $p_t$ towards $\ell$.
- The receptor generates a negative feedback $n_t$ if the receptor position $p_t \geq \beta$ the negative barrier.
- The negative feedback $n_t$ reverses the progress of the receptor position $p_t$
- The receptor position $p_t$ and negative feedback $n_t$ are updated according to a decay function

$$f_t : (p_t, n_t, u_t, \beta) \rightarrow (p_{t+1}, n_{t+1}) \tag{5.2}$$

where $p_{t+1}$ and $n_{t+1}$ are given in Eq. 5.3 and Eq. 5.4.

$$p_{t+1} = bp_t + u_t - an_t \tag{5.3}$$

$$n_{t+1} = \begin{cases} dn_t & \text{if } p_t < \beta \\ dn_t + g & \text{if } p_t \geq \beta \end{cases} \tag{5.4}$$

The parameters b and d are the receptor position decay rate and negative feedback decay rate with $0 < b < d < 1$. $a > 0$ controls the influence of negative feedback. $g > 0$ is the negative feedback growth rate.

- If the receptor position is on and above the receptor length $\ell$, then a classification occurs and the receptor is activated and considered anomalous ($c = 1$ if $p_t \geq \ell$, otherwise $c = 0$).

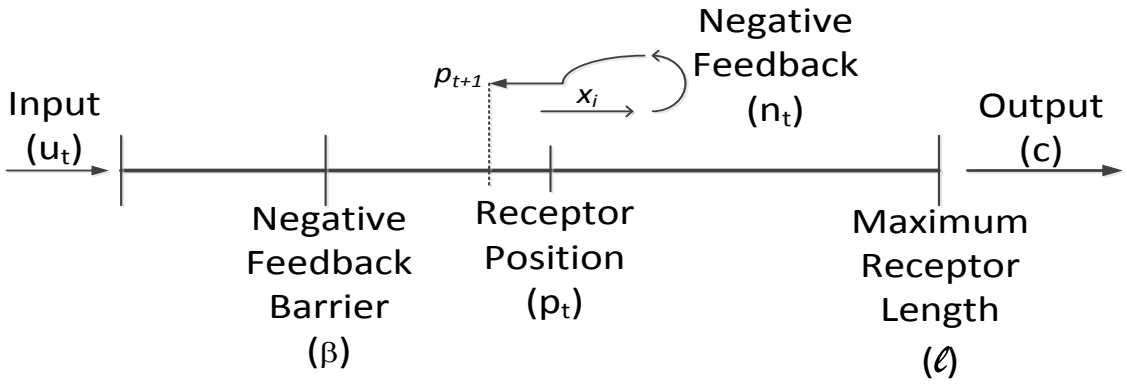



Figure 5.2: The Receptor from Owens et al. (2012): An input $u_t$ will move the receptor position to $p_t$ as well the generate a negative feedback if $p_t > \beta$. This negative signal reverses the receptor position to $p_(t+1)$. If the subsequent input signal is strong, the $p_t$ progresses further toward $\ell$ where the output *c* is produced when $p_t = \ell$.

## 5.3 IDRS: Interference Detection and Recovery Systems

In this section, we propose the application of RDA to diagnose and determine the cause of the failure using the current radio RSSI values collected within a time window. RDA is designed to detect any continuous time-series anomalous event and has the ability to adapt to dynamic changing environment by reshaping the normal pattern in the system when required. The RSSI has been recognised as a good predictor of link quality and has been used in routing protocol to assist in

detecting link failure (Srinivasan et al., 2010; Gnawali et al., 2009). Specifically, it has been shown that if the RSSI is higher than a sensitivity threshold [$RSSI_{th}(x)$] (about -87dBm), the RSSI correlates very well with the packet reception rate. This will allow us relate the distribution of the RSSI observed to the PDR generated by the MRP for detection and classification. However, the raw RSSI values consist of a mixture of different signal including noise. Any value less than -87dBm is unusable as it includes the background noise (Srinivasan et al., 2010). It is necessary to remove the noise in order to process and extract the interference pattern from the raw RSSI. Hence, we integrate the RDA into the MRP to provide an immune-inspired self-healing system that allow individual node to detect, determine and recover from failure online.

The objectives of the IDRS are:

1. To accurately identify the interference that is affecting the communication between a node and its neighbour in a distributed manner,

2. To make autonomous decisions on the recovery action to mitigate the effect of the interference, and improve the network reliability and efficiency.

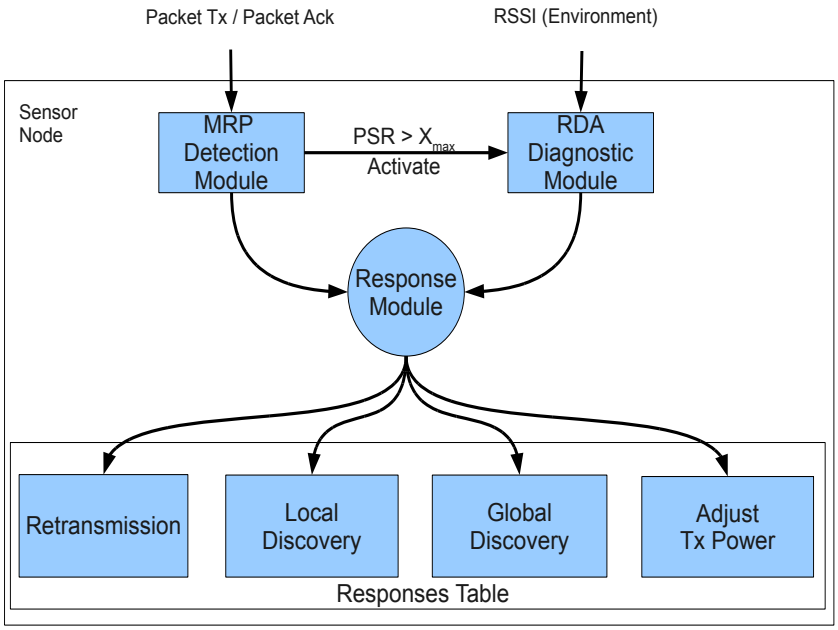



Figure 5.3: The architecture of the Interference Detection and Recovery System.

The IDRS in Figure 5.3 consists of three modules, representing each stage in the CIS: MRP Detection Module (MDM), RDA Diagnostic Module (RDM), and Radio Interference Response Module (RIRM). Inputs to the IDRS are the PSR and the RSSI. These inputs can easily be obtained and calculated from the node. The

MDM acts as the first line of defence to provide initial detection and response to the interference. If the condition does not improve, the MDM will activate the RDM to identify the type of interference based on the RSSI. Based on the results from both the MDM and the RDM, the RIRM will activate one or a combination of responses. By using the close feedback loop provided by the link layer, the effectiveness of the responses can be evaluated by the MRP. The cost of each response can be adjusted accordingly. Hence, IDRS should be able to recognise and respond to the failure based on the strength and duration of the interference.

In the following subsections, a detailed description of the proposed IDRS Algorithm is presented.

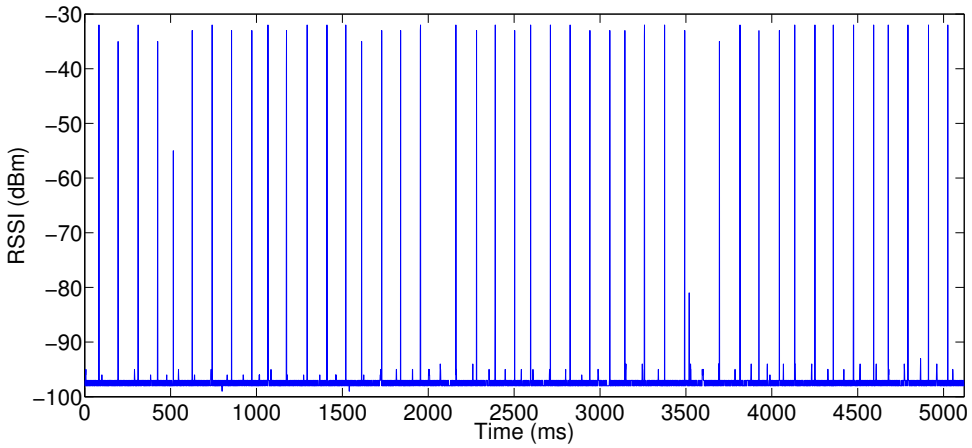
### 5.3.1 MDM: The MRP Detection Module

In WSNs, the packet reception ratio (PRR) is commonly used as a metric to detect network anomalies. The PRR is shared between neighbouring nodes (Lin et al., 2006). This data is usually piggybacked on an existing packet. However, in the presence of interference this data may be lost or corrupted. Hence, we advocate that the detection module should be implemented at the transmitting node. In the IDRS, we propose the use of the MRP to detect the presence of interference based on the PSR to provide an initial response. The PSR is the total number of packets successfully sent over the total number of attempts made in a given time window. The MRP detects deviation in the PSR and utilises the packet acknowledgement ($P_{ack}$) to provide initial network recovery response and activate the RDM. Each route recovery response incurs a specific cost ($RT_{cost}$ for *retransmission*, $LD_{cost}$ for *local recovery*). Associated with each recovery response is a maximum cost threshold: $RT_{max}$ for retransmission and $LD_{max}$ for recovery. The recovery response will only be selected if the cost of carrying out the response is lower than the maximum threshold. All these responses utilise the existing acknowledgement mechanisms on the link layer. As such, no additional communication overhead is incurred in the network.

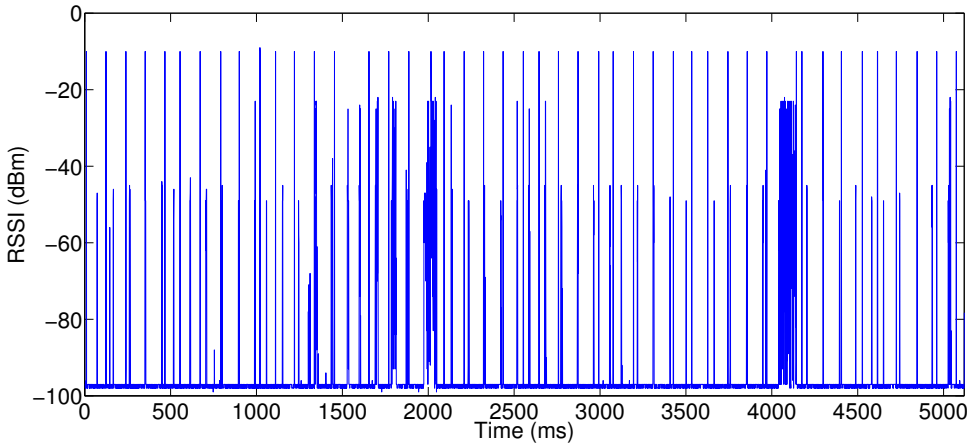
### 5.3.2 RDM: The RDA Diagnostic Module

To identify the cause of a transmission failure, the RSSI is used. Monitoring the RSSI in WSNs has been widely used to decide the required transmission power to transmit a packet (Boers et al., 2010). However, as illustrated in Figure 5.4, the RSSI values are sensitive to changes in environment and to classify the RSSI using traditional statistical techniques to differentiate fluctuating RSSI pattern is challenging (Boers et al., 2010). The RSSI values consist of a mixture of gaussian

(background) noise, packet signal and radio interfering signal. Small changes in the operating environment can trigger large variations in the RSSI due to the presence of noise, making it difficult to determine the type of interferences (Ko and Terzis, 2010). We propose the use of the RDA (Lau et al., 2011; Owens et al., 2009) to remove the background noise and classify different types of interference. The RDA has ability to achieve high positive detection rate and low false detection rate. Its ability to recognise anomalies in a dynamic environment has motivated its application to our solution.




(a) Normal RSSI Radio Data Pattern




(b) Abnormal RSSI Radio Interference Pattern

Figure 5.4: The raw RSSI data collected from the radio interface of a TelosB for both normal (Figure 5.4(a)) and abnormal (Figure5.4(a))

In order to generate the normal receptor signature on a node, the white background noise needs to be removed from the signature to reduce the number of false positives. These noises are determined by capturing the RSSI values from the radio transceiver without data communication. Based on RSSI values cap-

tured, the RSSI values range from -100 to -88 dB are known to be noise (Srinivasan et al., 2010). Based on these values, any RSSI values less than -87 dB can be discarded as its receptor's position will always exceed the threshold ($\ell$) as shown in Figure 5.5(a). Once the receptors representing the background noise have been identified and removed, the actual data signature can be generated using the RSSI values captured during normal data transmission without the interference.

To apply the RDA, the RSSI input data is divided into $s$ discretised locations and a *receptor* $\mathbf{x}_s$ is placed at each of these locations. A receptor has a maximum length $\ell = S_{max}(x)$, a position $r_p \in [0, \ell]$, and a negative feedback $r_n \in (0, \ell)$. The maximum receptor length $\ell$ (activation threshold) is set based on the spread of the RSSI value obtained during training. At each time step $t$, each receptor takes input $\mathbf{x}_i$ and performs a binary classification $c_t \in$ 0,1 to determine whether that location is considered anomalous. The classification decision is determined by the dynamics of $r_p$ and negative feedback $r_n \in (0, \ell)$.

The processes for initialisation and classification of the RSSI values are described as follows:

**Phase 1: Initialisation**

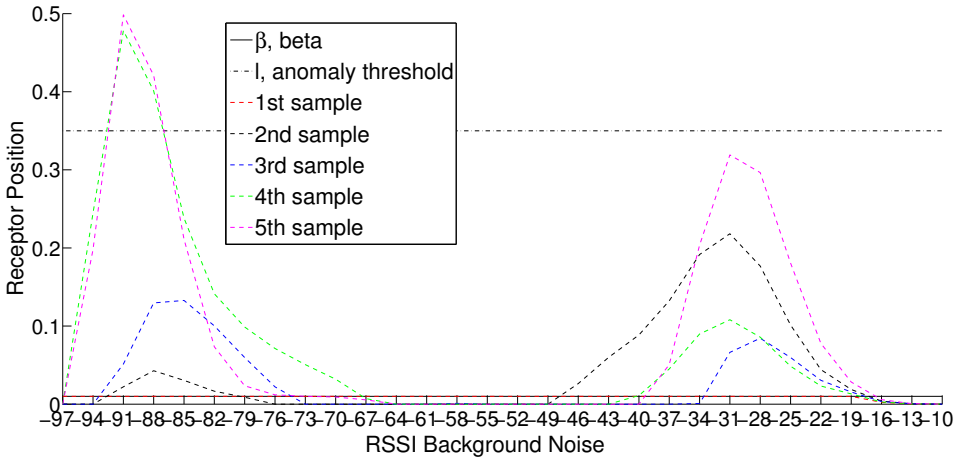
1. Present the normal RSSI values **X** (Figure 5.4(a)) to the RDA to generate its normal signature (Figure 5.5(a)). For each receptor $x$ , calculate the sum of stimulation $S(x)$ on each receptor $x$ for each RSSI input $x_i$, $x_i \in$ **X** .

$$S(x) = \sum_{i=1}^{n} \frac{e^{\frac{-(x-x_i)^2}{2h^2}}}{h\sqrt{2\pi}} \tag{5.5}$$

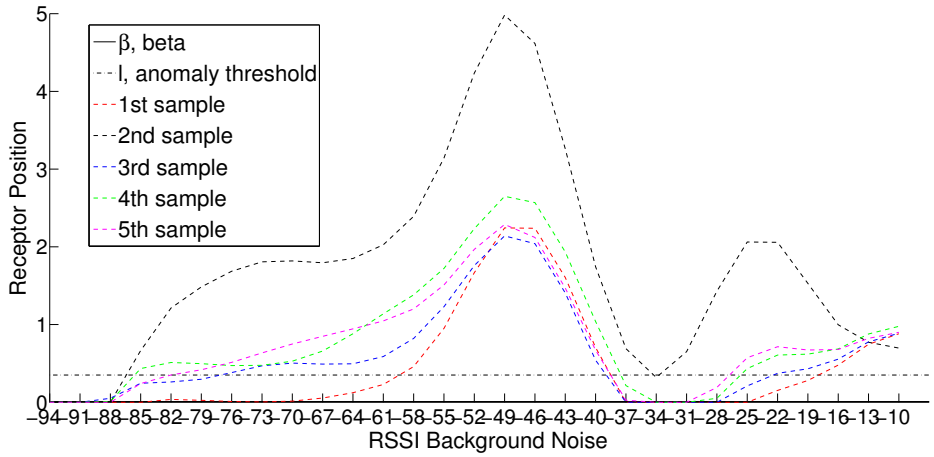
   where $h$ is the kernel width and $n$ is the total number of normal RSSI values.

   It is necessary to use appropriate kernel width to accurately detect the interference. To set the $h$, three sets of training data (normal, www and video streaming) are used to analyse the effect of $h$ on the detection rate. From figure 1, a small $h = 1$ pushes the receptor position far away from the threshold that may increase false positive detection (mis-classification). A large $h = 10$ may decrease the true negative rate (missed detection) as the reception position is below the threshold. By manually adjusting the value of $h$ using the training data, the RDA can yield a high true positive rate when $h = 5$.

2. Calculating the negative feedback $r_n(x)$ for each receptor $x$. $r_n(x)$ slows down the progression of the receptor position to reduce the false posi-




(a) The signature of activated receptor for the normal RSSI data Pattern




(b) The signature of activated receptor for the abnormal RSSI interference Pattern

Figure 5.5: The raw RSSI data shown in Figure 5.4 are fed into RDA to produce normal 5.5(a) and abnormal 5.5(b) signatures of activated receptors.

tive rate and is computed based on the base negative barrier $\beta$. Hence, it is necessary to determine the value of $\beta$ by adjusting the $\beta$ using a set of training data. From Figure 5.3.2, the position of the receptors moves toward threshold $\ell$ faster if $\beta$ is high ($\beta$ =1). The progression of the receptor position decreases when $\beta$ is small ($\beta$ =0.01). By manually inspect the receptor position with different value of $\beta$, $\beta$ is set a low value ($\beta$ =0.01) to yield low false positive rate.

$$r_n(x) = \begin{cases} S(x) - \beta, & \text{if } S(x) \geq \beta \\ 0, & \text{otherwise} \end{cases} \tag{5.6}$$

**Phase 2: Classification**

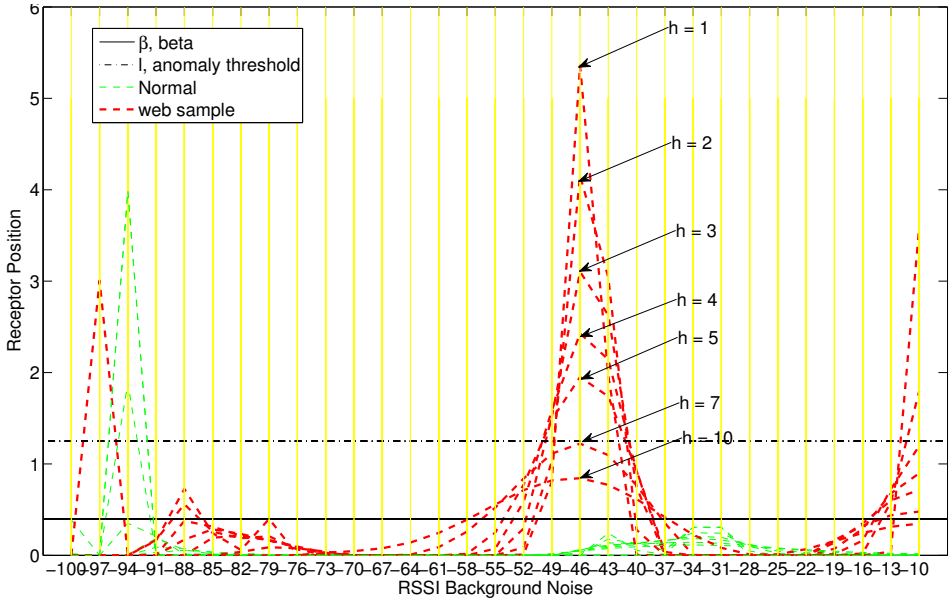



Figure 5.6: The signature generated by RDA with different values h. The receptor position progress faster if $h$ is high and may increase the false positive rate.

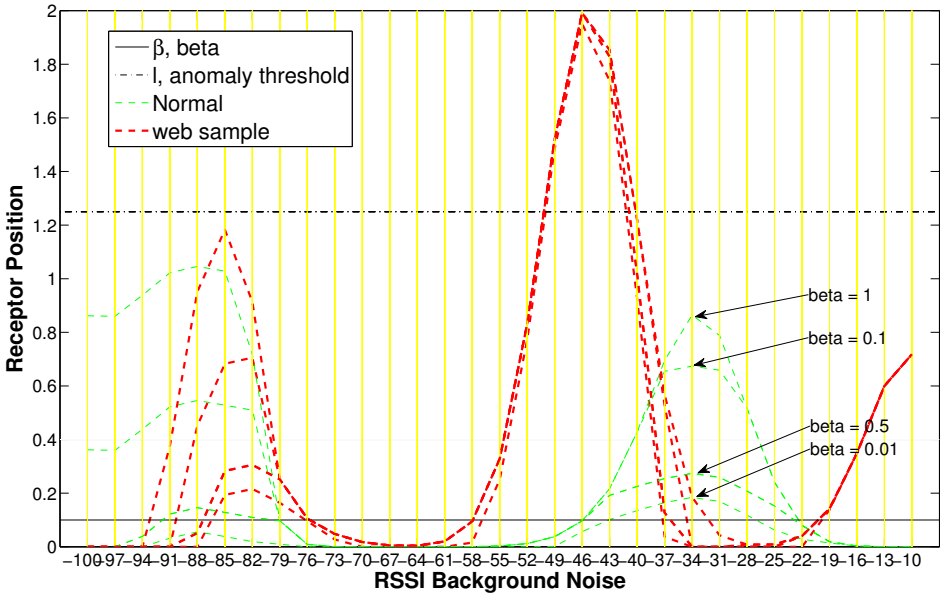



Figure 5.7: The signature patterns generated by RDA with the different values $\beta$. A low of value $\beta$ slow down the progress of the receptor position and hence may reduce the false positive rate.

1. Initialise the receptor position $r_t(x)$=0 for all receptors.
2. Based on the MAX($r_p(x)$) of normal signature, set the threshold value

of the receptor length $\ell = S_{max}(x)$. $S_{max}(x)$ is the maximum peak of the normal signature generated over 100 runs during Phase 1.

3. Calculate the new receptor position $r_p(x)$ with current RSSI values $\mathbf{V}$.

$$K_s = \sum_{i=1}^{n} \frac{e^{\frac{-(x-v_i)^2}{2h^2}}}{h\sqrt{2\pi}}, \qquad r_p(x) = K_s - r_n(x) \tag{5.7}$$

where each RSSI value $v_i \in \mathbf{V}$.

4. Classify $\mathbf{V}$:
A receptor is activated when

$$\mathbf{V} = \begin{cases} Normal, & \text{if } r_p(x) < l \\ Interference, & \text{otherwise.} \end{cases} \tag{5.8}$$

The classification of $v$ to different classes of interference is based on two variables (Figure 5.8$c$):

- The difference between distance of the highest receptor position and $\ell$ (max($\ell - r_p(x)$)), referred to as *Intensity*;
- The number of activated receptors, referred to as *Duration*.

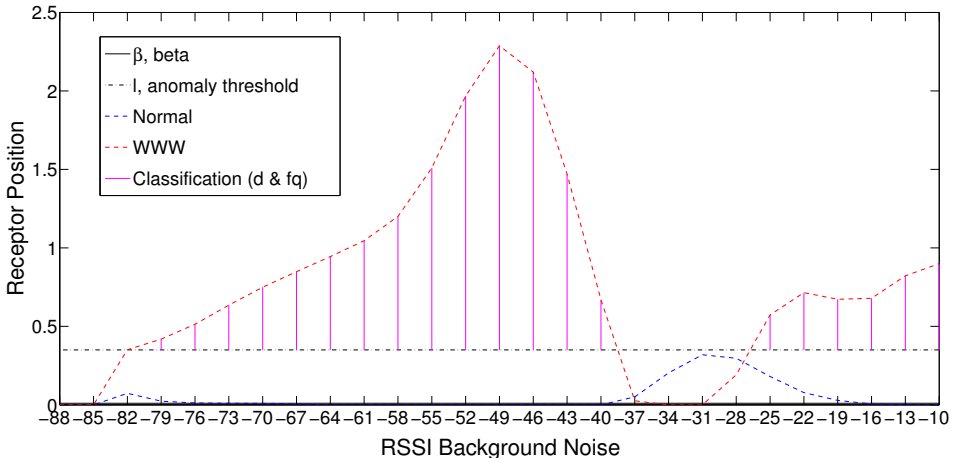



Figure 5.8: Using the outputs generated by RDA, the interference can be classified into either Class I, II, or III based on the euclidean distance of the furthest activated receptor and the number of activated receptors above the threshold $l$, represented by the global maximum and vertical lines in $(c)$ respectively.

### 5.3.3 RIRM: Radio Interference Response Module

According to the duration and strength of the interference introduced, we classify the interferences into three patterns namely: short and weak, medium, long and strong. Three different responses to overcome the interferences under investigation have been associated to each of the patterns classified namely:

1. Weak and short: Retransmission (RT) is the optimistic approach to overcome transient failure. It is activated when the MDM detects that the acknowledgement packet $P_{ack}$ has timeout and the cost of retransmission ($RT_{cost}$) has not exceed the threshold ($RT_{max}$). This response is particularly effective when the network is suffering from frequent intermittent packet collisions created by other radio emitting devices with a stronger signal (Gutierrez et al., 2001). Increasing the transmission stronger than the interference source may also help to handle unavailable route caused by an obstacle or a weak interference source. It is a common to increase the transmission power to penetrate through the obstacle in order to communicate with the next hop neighbouring node (Boers et al., 2010; Lin et al., 2006). However, the use of higher transmission power can only be applied if the interference source is known to be weaker than the node's transmission signal as a higher radio transmission consumes more battery and may also interfere with other nodes.

2. Medium: Local discovery (LD) is activated when the node failed to send the packet after several RTs (as indicated by PSR) or when the RDM identified an interference that has *medium* strength and duration. As a medium interference may only affect one of two nodes, other local nodes within the affected area can still forward the packet to the next two hop neighbour. This response is also best executed when the next node is permanently unavailable.

3. Strong and long: Global Discovery (GD) is usually the last option to take when the local nodes are spatially interfered and the existing route is known to be unreliable. This action is usually taken when all the previous responses have failed, and there is no local node available to re-route the traffic. This type of failure is usually created by the interference source that is *long* and *strong* affecting all the nodes.

Beside the three interference patterns presented above, there are other interference patterns that are not identified by our classification algorithm. To identify

other interference patterns such as strong intensity and short duration, each interference needs to be generated and captured on the node before it is processed and classified by the RDA. Once the new class has been determined, an appropriate routing protocol can be mapped onto the new class.

A decision tree, based on expert knowledge, is presented in Figure 5.9 to show the response strategy to be selected based on the current network environment. The algorithm for IDRS with MRP, RDM and RIRM is given in Algorithm 2.

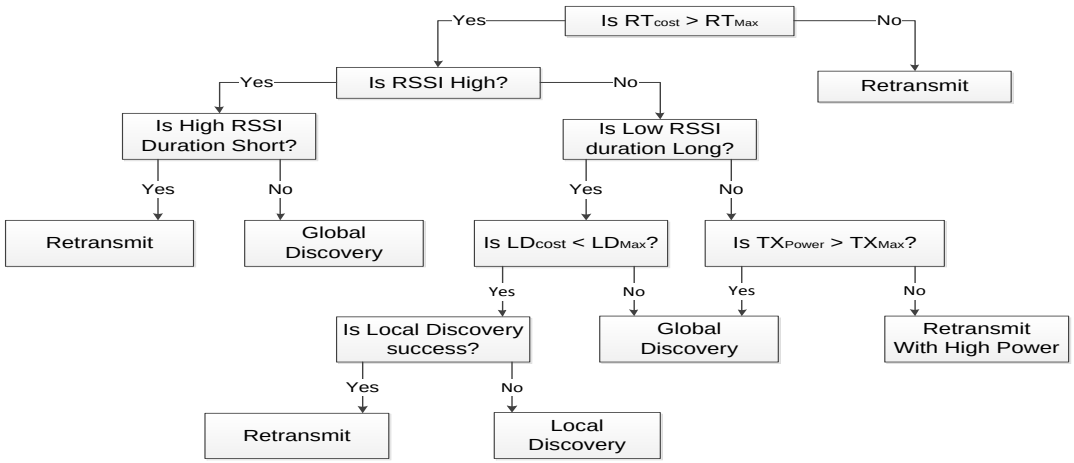



Figure 5.9: Decision tree based to respond to different interference.

## 5.4 Experiments and Results

We conducted two experiments to evaluate the proposed IDRS. The first experiment examines the effectiveness of the RDA classifier in the RDM. The second experiment evaluates the efficacy of the proposed IDRS when compared to other methods.

### 5.4.1 Evaluation of the RDM

In the RDM, we use over 850,000 RSSI readings to classify the interference introduced into three classes: CLASS I, CLASS II, and CLASS III. The RSSI values are obtained from the TelosB radio module, exposed to different interference sources with weak, medium and strong intensity. The spectrum of the RSSI values used is the range of -100dBm to -10dBm. Due to the limited processing power and storage in a sensor node, this spectrum is uniformly divided into 30 slots to ensure that each RSSI values are evaluated. The RDA requires $O(n^2)$ processing overhead, hence will only activate if PDR $<$ N (where N is the minimum PDR the application can afford to tolerate). A receptor is used to represent each slot.

**Algorithm 2:** IDRS Algorithm with the combination of MRP and RDA

**Input :** Packet Send $P_s$
**Output:** Response Action

```
while Packet Buffer is not Empty do
    Send Packet P_s and wait for acknowledgement P_ack
    if P_ack is not received then                                   } Activate MRM
        Calculate Packet Sending Ratio, PSR                         } Detection
    else                                                            } Module.
        Decrease Retransmission Cost, RT_cost
    if PSR < 95% then                                               } Activate RDA
        Determine interference CLASS from RDM                       } Diagnostic
    if not CLASS III and [PSR > 90% or RT_cost < RT_max] and         } Module.
    Route is valid then                                             } Trigger
        Retransmit                                                  } RT Response.
        Increase Retransmission Cost, RT_cost                       }
    else if CLASS II and LD_cost < LD_max then                      } Trigger
        Perform Route Discovery                                     } LD Response.
        Increase Local Discovery Cost LD_cost                       }
        if Route Discovery is Successful then                       }
            Decrease Retransmission Cost RT_cost                    }
    else if CLASS I and Tx_Power < Tx_MAX then                      } Trigger Higher
        Increase Transmission power, Tx_power                       } Transmission
        Decrease Retransmission Cost, RT_cost                       } Power Response.
    else
        Invalidate Route and Send Error                             } Trigger
        Global Discovery                                            } GD Response.
    if Timeout then
        Reinitialised
```

In order to classify the interference into different classes, the pre-processed training data from the RDA have to be grouped according the number of activations (duration) and the maximum distance between $\ell$ and maximum receptor position. A scatter plot is used to investigate whether a general pattern can be observed from the preprocessed data. From Figure 5.10, the output generated by the RDA can be grouped into 3 different classes based on different *Intensity* (**C1**) and *Duration* (**C2**) representing:

- **Class I**:- Weak intensity, short duration ($C_1 < i_1$ and $C_2 < d_1$)
- **Class II**:- Medium intensity, medium duration ($i_1 < C_1 < i_2$ and $d_1 < C_2 < d_2$)
- **Class III**:- Strong intensity, long duration ($C_1 > i_2$ and $C_2 > d_2$)

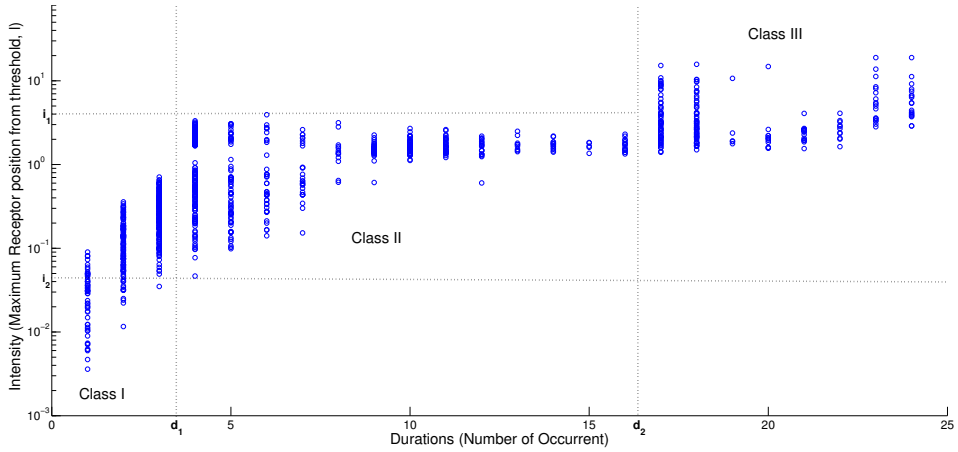



Figure 5.10: The distribution of the interference characteristics after processed by the RDA. The distribution can be grouped as the three different interference classes (Class I, II and II) as separated by the dotted lines

Hence, an unsupervised K-mean clustering algorithm is applied to data group the data into 3 groups. K-means clustering is a method used in data mining to partition $v$ samples into $c$ clusters in which each sample $v$ belongs to the cluster with the nearest mean. It is performed offline as it is computationally expensive. The classes do not usually change throughout the node's lifetime unless it is deployed in a different physical location or moved. The derived classes based on C1 and C2 is shown in Table 5.1.

**Evaluation Metrics:**

We evaluate the performance of the RDM in TelosB mote based on *sensitivity* (Equation 5.9) and *precision* (Equation 5.10).

Table 5.1: Interference Class based on the intensity and duration of the interference experienced by a node

| **Intensity, C1** | **Duration, C2** | Class | Remarks |
|---|---|---|---|
| $0 < \mathbf{C1} \leq 2.8$ | $0 < \mathbf{C2} \leq 5$ | I | weak intensity, short duration |
| $2.8 < \mathbf{C1} \leq 11.0$ | $5 < \mathbf{C2} \leq 16$ | II | weak intensity, long duration |
| $\mathbf{C1} > 11.0$ | $\mathbf{C2} > 16$ | III | strong intensity, long duration |

$$Sensitivity = \frac{TP}{TP + FN} \quad (5.9) \qquad Precision = \frac{TP}{TP + FP} \quad (5.10)$$

*Sensitivity* measures how well the RDM can correctly classify the interference source whilst *precision* measures the probability of a detected event that is representing a true positive result rather than a false positive. *Precision* ensures the appropriate response is taken for the corresponding interference.

**Experimental Setup:**

Two static nodes are deployed so that they are within each other transmission range. One node is configured to transmit packets at the rate of 8 packets per second to simulate heavy load traffic while sampling its radio channel at the rate of 1 KHz to collect the RSSI values and perform online detection.

Seven different network conditions, each representing different interferences commonly occur an office and home environment, are used to test the system namely: object blocking, jamming, web browsing, slow streaming, fast streaming, slow downloading, and fast downloading (torrent). The descriptions of each of the traffics are provided in Table 5.2. In each run, the interference is injected into the network at periodic interval to capture the PSR affected by the interference. This is done by placing a laptop next to the receiving node. Due to the limited memory size in the nodes to store the log, each experiment is run for 5 minutes to generate 2400 packets and is repeated 15 times.

**Experimental Results:**

The results for the experiment are shown in Figure 5.11. From the figure, the RDM has achieved a precision of above 80% when the interferences have caused a drastic drop in the PSR $< 70\%$. The occurrence of these two interferences requires an alternative route to deliver the packet successfully. Hence, it is important these two interferences are correctly identified to avoid unnecessary responses to be executed. With RDM, a precision rate from 80% to 90% has been achieved for both blocking and fast download. Although the RDM can only classify 50-60%

Table 5.2: Descriptions of different type of traffics generated using the laptop.

| Traffic | Descriptions |
|---|---|
| Object blocking | People continuously walking across the communication path between the two nodes. |
| Jamming | A local node continously transmitting packets placed near the sending node. |
| Web browsing | Casual surfing of websites (searching in google.com, bbc.co.uk and soccernet.com). |
| Slow streaming | Watching video streaming from the news channel in Brunei (http://www.rtbnews.rtb.gov.bn/). |
| Fast streaming | Watching multiple HD video streamed within uk like BBC iplayer. |
| Slow downloading | Downloading file from file-sharing website with an average speed of speed of 518Kbps. |
| Fast downloading | Downloading file using file sharing software (torrent) with average speed to 2Mbps. |

of the class II interference, its impact on the network PSR is less extreme with more than 80% of the packet is still being delivered compare to blocking and fast download with the PSR below 70%. Beside, high accuracy in Class II interference is usually not required for accurate response as the sequential recovery step (Retransmission followed by local discovery) provided by the MRP can usually overcome weak interference and rectify the problem.

### 5.4.2 Evaluation of the IDRS

In this second experiment to compare the efficacy of the IDRS to NST, MRP, and MRP with adaptive transmission power, two sets of experiments are executed: one in hardware and the other in simulation. The objective of the hardware experiment it to show that the IDRS can increase the PDR and reduce the communication overhead in the network. The simulation evaluate the scalability of the IDRS as it is much easier to test a large scale network in simulation than real hardware deployment.

To evaluate the performance of the IDRS against the NST, MRP, and MRP with adaptive TPC (MTPC), the routing protocols are installed and tested in both hardware and simulation. The MTPC protocol is used to evaluate the benefit of boosting the transmission power when the receiver is being blocked. The performance of the routing protocols is evaluated based on the following metrics:

- **Packet Delivery Ratio (PDR)**: PDR represent the percentage of the number of packets received by the receiver, to the total number of packets transmitted by the sender. This metric measures the reliability of the routing protocol.

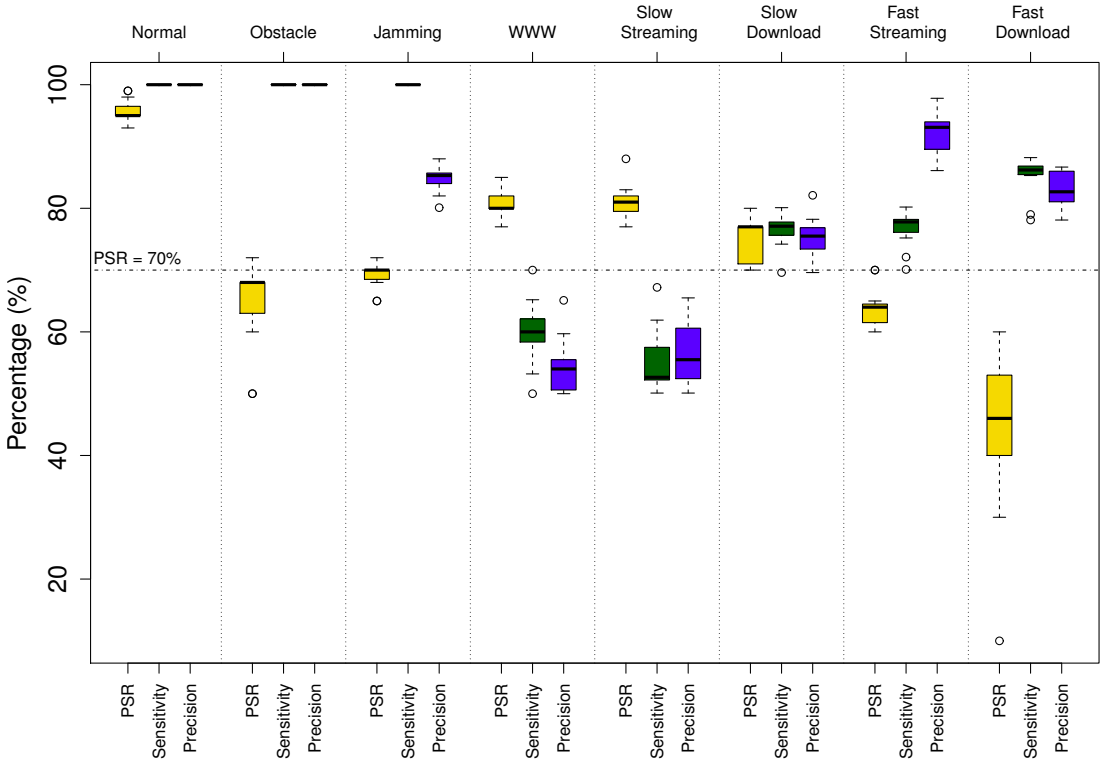



Figure 5.11: Results showing the detection accuracy and the effect of different interference classes on PSR. The RDM can detect and identify interference that have a severe impact on the PSR (PSR<70%) as above 80% precision can be achieved

- **Transmission Overhead (TO)**: TO is defined as the average number of transmissions made by a node to deliver the packets to the receiver. This metric represents the efficiency of the routing. It can be calculated by dividing the sum of the transmissions made, including RT, LD and GD, to the total number of packets received.

**Hardware Experimental Setup:**

Six static TelosB motes are placed 3 metres apart using the topology shown in Figure 5.1 to ensure that the neighbouring nodes are within each other transmission range. The experiment is conducted at the centre of a room relatively free from uncontrolled radio sources to ensure its correctness and validity. The node transmission is set to minimum power using the same channel as the WLAN in the room. A notebook with different applications will be used as an interference source. The LLN is enabled to allow packet acknowledgement in each node. During initialisation, node 2 is configured to collect temperature reading from the sensor and transmit the packet to node 3, at regular intervals (250ms) via the intermediate nodes. Once the network route has been established, and the normal

signature has been collected by the RDM (after 30 seconds), different interference sources are introduced into the network (close to node 5 and 6) at every *i* seconds intervals. Each interference lasts for approximately *d* seconds. In our experiment, the values of *i* and *d* are set accordingly ($i = 30$ and $d = 15$) to ensure that the networks can recover before the next interference source is injected. The PDR and the response executed are computed and logged by node 7 during each interference cycle. Due to the limited memory size to store the logs, each experiment is run for 10 minutes to capture 20 interference sources. Each experiment is repeated 15 times as it takes a longer time to configure and run the experiment in hardware.

**Experimental Results:**

The results for this experiment are shown in Figure 5.12 and 5.13 with their respective statistical test values in Table 5.4 and Table 5.5. During normal condition, NST has the lowest PDR (median at 92.5%) and requires more transmissions than MRP, TPC and IDRS. The transmission overhead for IDRS is slightly lower than MRP and TPC despite having the same PDR (Rank-Sum $p$-value $< 0.005451$, Vargha-delaney $A$-value $>0.79778$).

When errors are introduced into the network, the performance improvement made by the RDA with MRP is significantly better than MRP as the PDRs for IDRS are always higher than MRP. (For Class I is 2% higher, Class II is 4% higher and Class III is 9% higher with $p$-value $<< 0.005$ and $A$-value $< 0.27$). MTPC has the highest PDR in Class I and Class III. Further analysis on the responses executed by the routing protocol in Table 5.3 has shown the number of TPCs in MTPC is higher than IDRS at class III interference. We believe by increasing the transmission power during interference may have improved the PDR in MTPC. From Figure 5.12, by increasing the transmission power during Class I interference has improved the PDR by 5% on average for MTPC compared to MRP and 2.5% compared to IDRS.

Although the boxplot in Figure 5.12 has shown a significantly higher PDR in MTPC than IDRS , IDRS has a lower number of packet transmissions compare to MTPC as shown in Figure 5.13. When Class I and III errors are introduced, the transmission overhead for IDRS is less than MTPC and the PDR of MTPC is 2% higher. From Table 5.3, we can observe that IDRS generates fewer packets than MTPC as an appropriate response can be executed by RIRM based on the diagnosis made by the RDM. For example, the IDRS performed less RT and TPC in Class II interference as the node was able to recognise the interference and performed

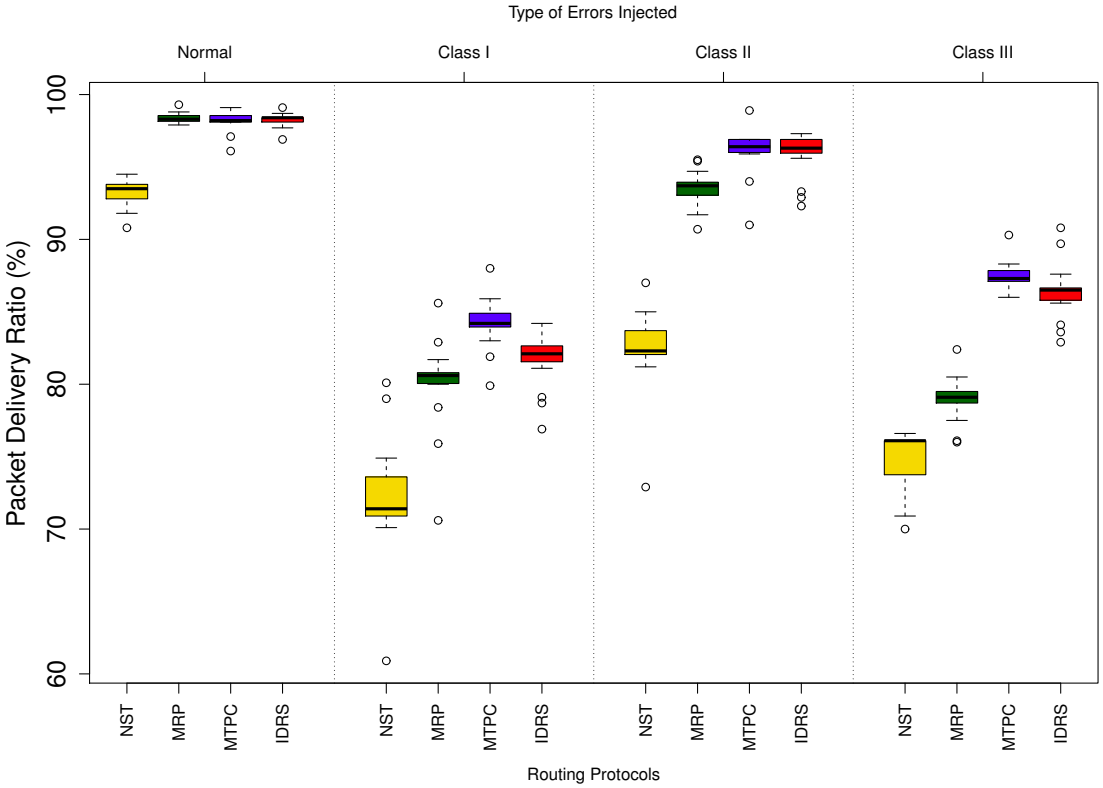



Figure 5.12: PDR achieves by different routing protocols for different classes of interference.

Table 5.3: The execution of different responses in all the nodes for IDRS and MTPC. Results show that IDRS can execute the appropriate response compared to MTPC. Class III interference has been correctly classified resulting in higher number of Global discovery and lower number of high power transmission.

| | Numbers of Each Response Executed | | | | | | | | Interference Detected | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | RT | | LD | | TPC | | GD | | | | |
| Class | IDRS | MTPC | IDRS | MTPC | IDRS | MTPC | IDRS | MTPC | I | II | III |
| Normal | 41 | 44 | 24 | 35 | 5 | 7 | 50 | 48 | 7 | 0 | 0 |
| I | *113* | *174* | *52* | *92* | *33* | *12* | *342* | *404* | **59** | 4 | 1 |
| II | 42 | 66 | 15 | 14 | 6 | 14 | 88 | 60 | 35 | **15** | 1 |
| III | **164** | **193** | **108** | **145** | **25** | **44** | **121** | **219** | 88 | 74 | **82** |

LD immediately (higher LD). The RDM in the IDRS managed to effectively classify the interference as shown in class I, II, and III in Table 5.3. As a result, the total number of responses performed by IDRS is significantly less than the MTPC. As a result, IDRS consumes less energy as a lower number of transmissions is required to deliver a packet successfully compare to NST, MRP and MTPC.

Table 5.4 and 5.5 also shows that the performance (PDR and TO) of IDRS is both statistically and scientifically significant different (in bold) from the other three protocols. Hence, the use of RDA to classify the radio signal noise pattern

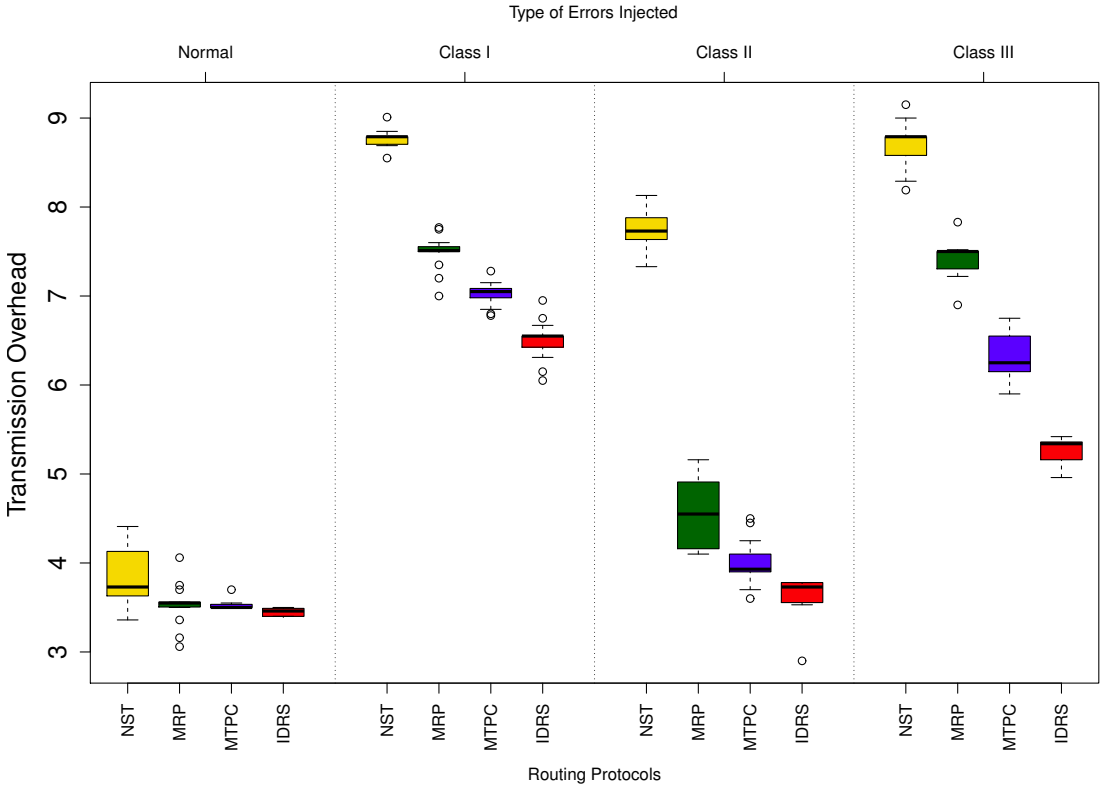



Figure 5.13: Transmission Overhead generated by different routing protocols for different interference classes of interferences.

has not only allowed the system to response accurately with minimal transmission overhead, but has also maintained a higher PDR.

Table 5.4: $p$-values (Rank sum test) to determine statistical significance of the performance between the routing protocols for the PDR and TO (Bold highlights significance value $p < 0.05$)

| Protocols | NST:MRP | MRP:MTPC | MRP:IDRS | MPTC:IDRS |
|---|---|---|---|---|
| | | *Packet Delivery Rate (%)* | | |
| *Normal* | **3.2011e-06** | 0.69072 | 1.00000 | 0.95004 |
| *Class I* | **8.3800e-05** | **0.00029** | **0.02595** | **0.00071** |
| *Class II* | **3.2994e-06** | **8.5659e-05** | **0.00102** | 0.80234 |
| *Class III* | **3.5431e-05** | **3.2581e-06** | **3.2994e-06** | **0.02223** |
| | | *Transmission Overhead* | | |
| *Normal* | **0.001941** | 0.129261 | **0.005451** | **4.8774e-06** |
| *Class I* | **2.9279e-06** | **2.8064e-05** | **2.9279e-06** | **5.7810e-06** |
| *Class II* | **3.3495e-06** | **8.5503e-05** | **3.1688e-06** | **9.1032e-05** |
| *Class III* | **2.9430e-06** | **3.1688e-06** | **3.0973e-06** | **3.2173e-06** |

Table 5.5: Vargha-delaney test to determine scientific significance of the performance between the routing protocols for PDR and TO. The *A*-values are computed (Bold highlights significance value)

| Protocols | NST:MRP | MRP:MTPC | MRP:IDRS | MPTC:IDRS |
|---|---|---|---|---|
| | | *Packet Delivery Rate (%)* | | |
| *Normal* | **0.00000** | 0.54444 | 0.50000 | 0.49111 |
| *Class I* | 0.07778 | **0.11111** | **0.26000** | **0.86444** |
| *Class II* | **0.00000** | **0.07778** | **0.14667** | 0.47111 |
| *Class III* | 0.05778 | **0.00000** | **0.00000** | **0.74667** |
| | | *Transmission Overhead* | | |
| *Normal* | **0.83333** | 0.66222 | **0.79778** | **0.98222** |
| *Class I* | **1.00000** | **0.94889** | **1.00000** | **0.98667** |
| *Class II* | **1.00000** | **0.92222** | **1.00000** | **0.92000** |
| *Class III* | **1.00000** | **1.00000** | **1.00000** | **1.00000** |

**Computation and Memory Footprint**

To measure the computation and memory footprint of IDRS, MRP, NST and AODV, we capture the number of CPU clock cycles required to send 50 packets using the Contiki's Cooja Simulator. Using the same Tiny-OS binary for the hardware, the number of CPU clock cycles processed by node 7 of Figure 5.1 is measured from the simulator. The statistics of the number of CPU clock cycles from 20 runs are given in Table 5.6 together with the memory footprint obtained during compilation.

Table 5.6: The memory footprint (in Byte) required for MRP and IDRS in a TelosB mote and the processor computational (Number of cycles) collected from a Contiki's Cooja Simulations

| Overhead | IDRS | MRP | NST | AODV |
|---|---|---|---|---|
| RAM (KBytes) | 5.12 | 4.01 | 4.00 | 3.99 |
| ROM (KBytes) | 34.33 | 32.20 | 32.04 | 31.66 |
| Number of CPU Clock Cycle (MCycles) | 61.7 0 | 58.08 | 59.17 | 54.69 |

From Table 5.6, the IDRS has the highest number of CPU clock cycles than the other routing protocols. The number of CPU clock cycles required to execute IDRS is 4.28% higher than MRP. However, the differences between them are not significant (p-value=0.056 a=0.3). The statistical test also shows no difference in the number of CPU clock cycles to send 50 packets between single mode (NST) and multi-mode (MRP) (p=value=0.21, a-value=0.385). The number

of CPU clock cycles utilises by MRP is 5.5% higher than AODV (p-value=0.0296 and a-value=0.30). However, Raghunathan et al. (2002) highlighted that the energy consumption required for processing is 70% less than communication. As a result, the increase in the processor footprint in the multimode routing is acceptable as the number of communications required for routing is reduced significantly, indirectly reducing the energy consumption in the nodes. The code and storage size required for the implementation of MRP and IDRS can fit in the existing mote.

**Simulation Setup:**

In order to test the scalability of the IDRS, the IDRS[1] is implemented and tested in NS-2 simulation and compare it against AODV, NST and MRP. MTPC is not evaluated in simulation because NS-2.34 does not support dynamic power control. To simulate the IDRS, the interferences experienced in the real networks need to be implemented in NS-2. Using the RSSI traces captured in the hardware experiment, these traces are preprocessed and transformed into a series of *on* and *off* patterns according to the values of RSSI (above -87dB is *off*, otherwise *on*). These *on* and *off* patterns will be used by the node to generate the failure. 50 different failures patterns for each class, each stored in a file are generated. Each of these files will be randomly selected by NS-2 to generate the failures observed by each node. In order for RDA to perform the diagnosis during failure, it is necessary to generate 100 sets of raw RSSI pattern for each interference class and store them separately in a file. These RSSI traces will also be randomly selected by the simulator during each failure detected by the MRP.

Two sets of networks are deployed. The first one is a small scale network based on the controlled hardware deployment of 6 nodes that mirrors the hardware experiment and its configurations defined in Section 5.4.2. In the second experiment, we employ the 51 nodes indoor deployment of Figure 4.3 of Section 4.3.2 to evaluate the IDRS in a larger network. To generate the failure for the second experiment, each node 20, 21, 22, 23, 24, and 25 will turn itself on and off according to the on-off pattern observed from the randomly selected failure file and node 0, 1 ,2 and 3 will transmit periodically every 0.2s. As it is faster and easier to run test the routing protocol in simulation, each simulation is repeated 50 times.

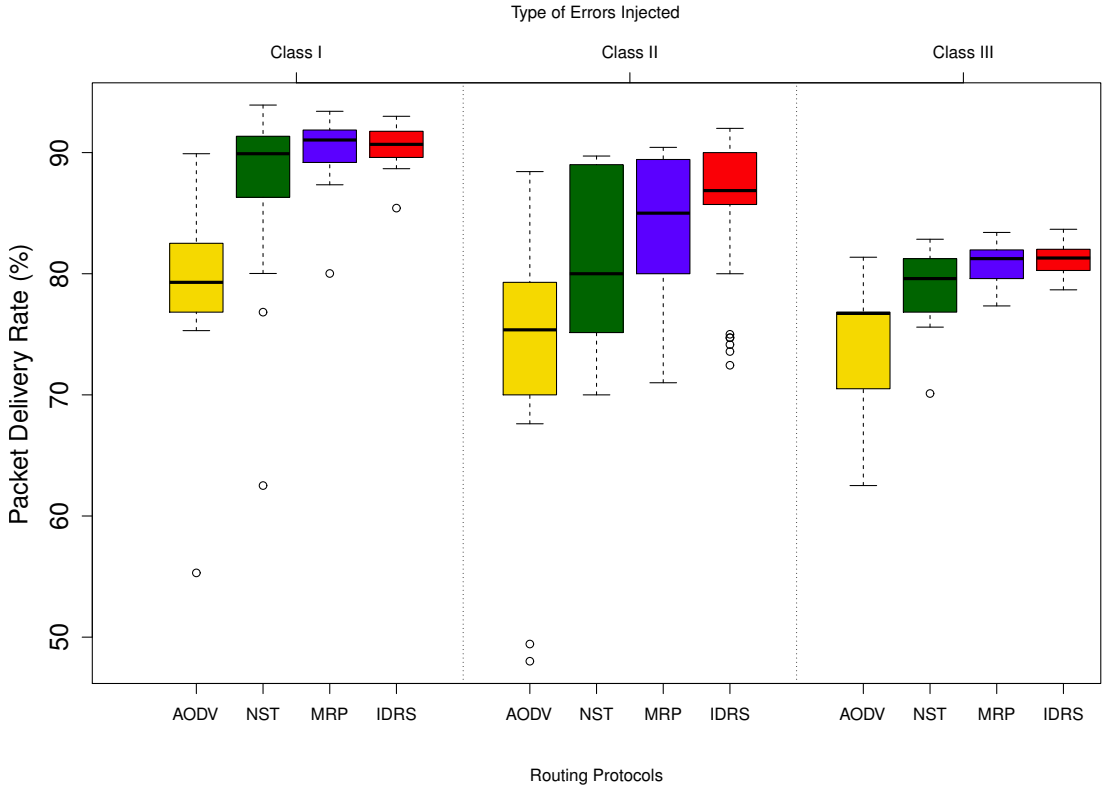
[1]Downloadable at http://rtslab.wikispaces.com/file/view/idrs.tar

**Simulation Result:**

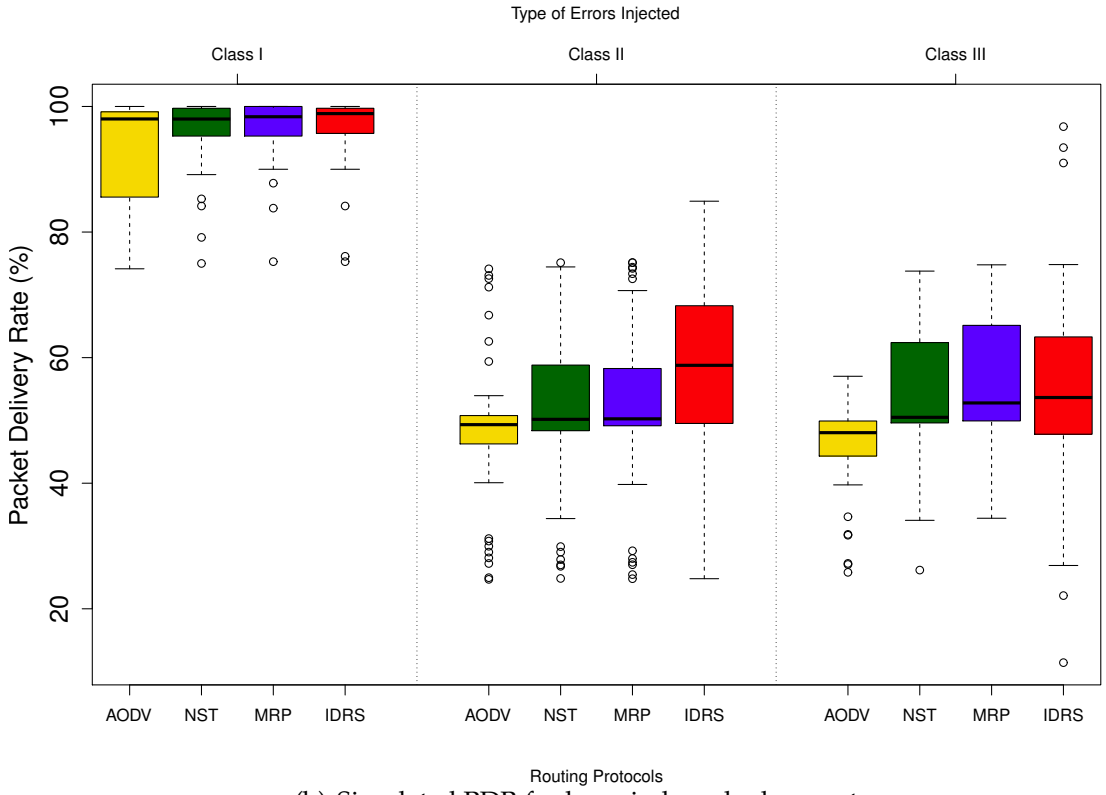
The four performance metrics defined in Section 3.1.5 are used to evaluate the results generated by NS-2.34 simulator. Figures 5.14, 5.15, 5.16 and 5.17 show the performance of the routing protocol induced with different interferences. The overall results have shown that the performance of the IDRS is not affected by the size of the network as shown by the PDR (Figure 5.14) of the small controlled network and large indoor network. However, the communication overhead generated by IDRS is significantly less than MRP, NST and AODV in both small and large networks (Figure 5.15) when induced with a Class III interference compared to Class I and II interferences (Medium $A$-value $> 0.67$). The effect on Class I interference on the routing protocols produced in simulation is not significant as shown by both the $p$ ($> 0.05$) and $A$-value ($< 0.73$ or $> 0.27$ ) in Table 5.7. The boxplot also shows different trends between the hardware and simulations.

In the small controlled environment, the PDR of both MRP and IDRS is significantly higher than NST and AODV (above 90%) in Figure 5.14(a). Although there are no differences between the PDR for IDRS and MRP, significantly lower routing overhead is observed for IDRS in Figure 5.15(a) for Class III interference with the p-value $\leq 0.01$ in Table 5.7 with scientifically significance $A$-value. Each time the RDA detects the class III interference, the MRP is alerted. As a result, local repair is avoided; the amount of routing overhead and energy consumption are lowered. Although there is no significant difference in the energy consumption in *ENG* column of Table 5.7 between the MRP and IDRS, the highest energy consumed by IDRS during Class III interference is 10% less as shown in Figure 5.16. The time to deliver the packet in IDRS is also significantly small compared to NST and AODV. This reduces the MTTR in Equation 5.1 and increases the availability.

In the large indoor environment, more packets have been delivered by IDRS than AODV in class II and III interference with $p$-value $< 0.001$ and $A$-value $< 0.3$. The variability of PDR observed in IDRS is significantly larger than the other three protocols as shown in Figure 5.14(b). This is due to the time varying behaviour, and the flexibility and different options available in IDRS to switch between different routing protocols and respond to the failure over a periods of time. Some responses produce high PDR and less routing overhead as shown in Figure 5.15(b). While in other case, the response gives the same minimum PDR as the other routing protocol. Similar to the control environment, the routing overhead generated during class III interference is also significantly lower in IDRS with $p$-value $< 0.006$ and $A$-value $< 0.67$. Hence, the results from the simulations have shown that by using the RDA to identify the type of interferences in the

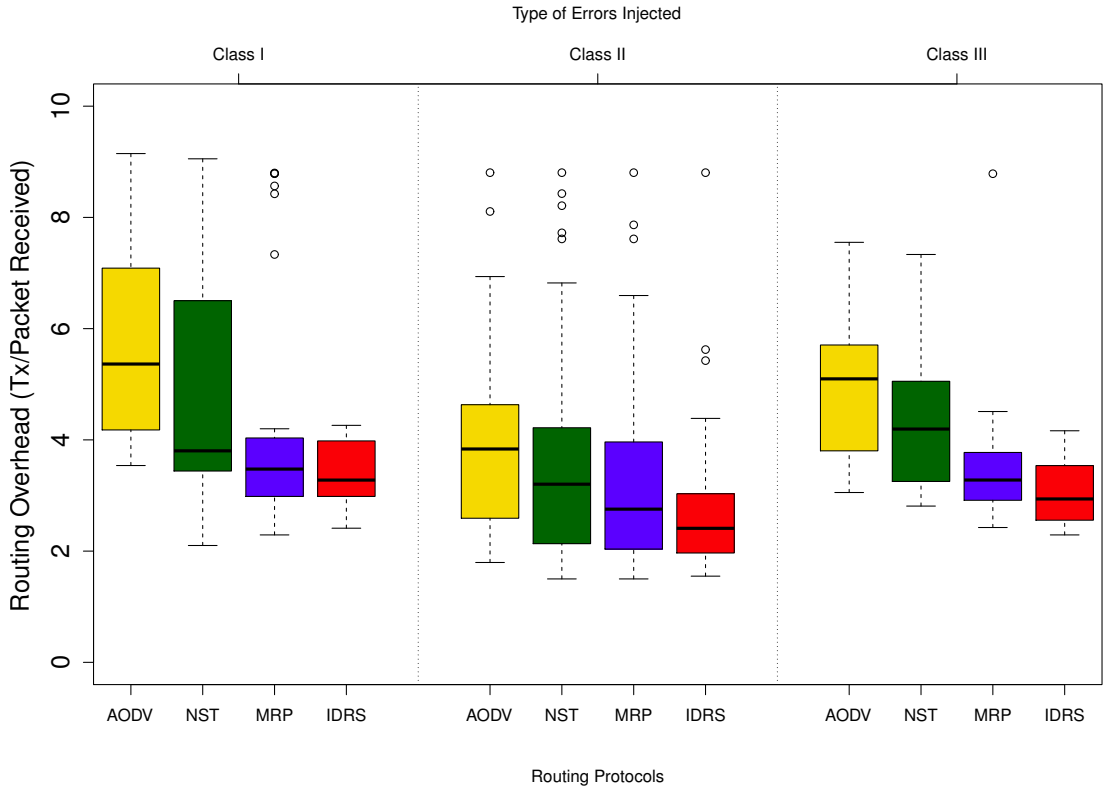



(a) Simulated PDR for small controlled environment

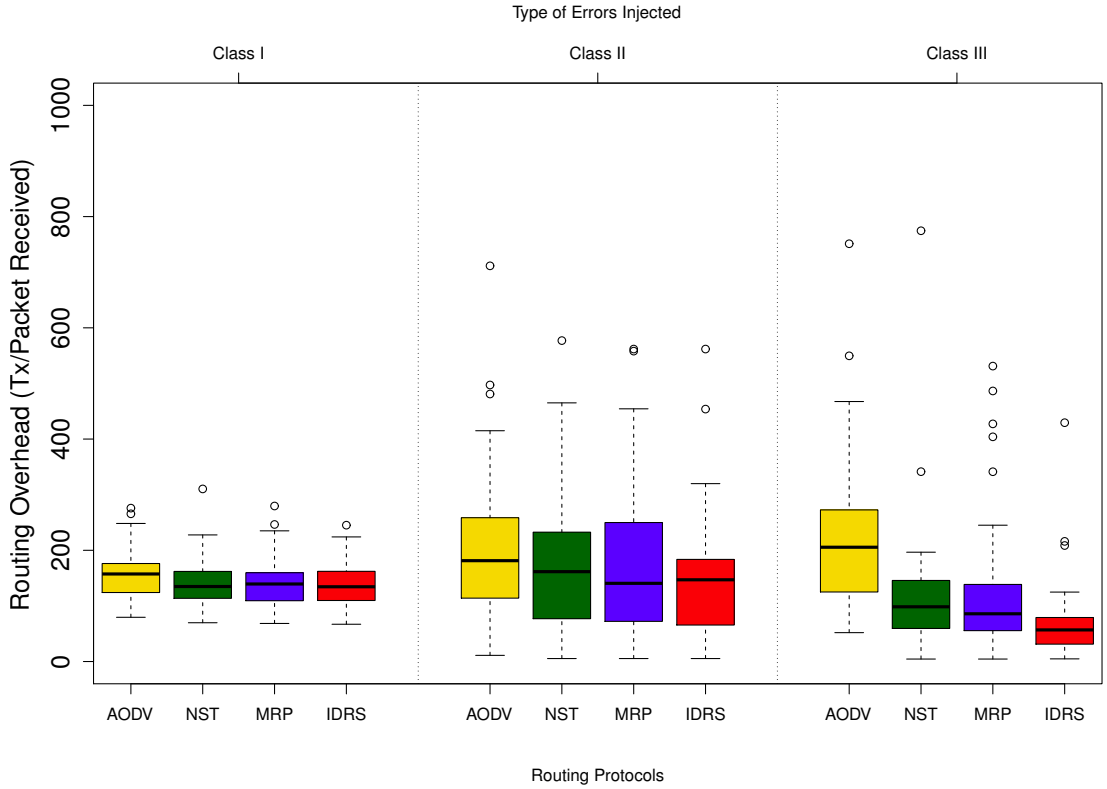



(b) Simulated PDR for large indoor deployment

Figure 5.14: The Box Whisker plot showing the packet delivery ratio achieved by the AODV, NST, MRP and IDRS for both small controlled 5.14(a) and large indoor 5.14(b) environments induced with the 3 different interference classes.

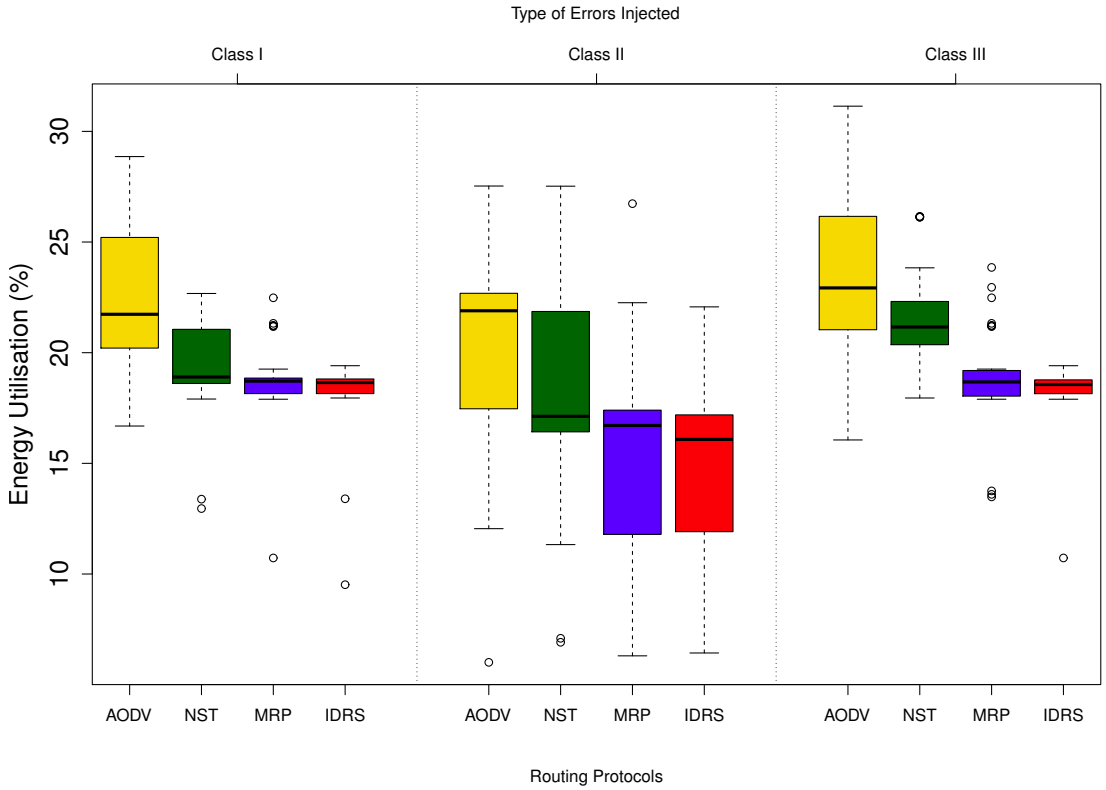



(a) Simulated routing overhead for small controlled environment

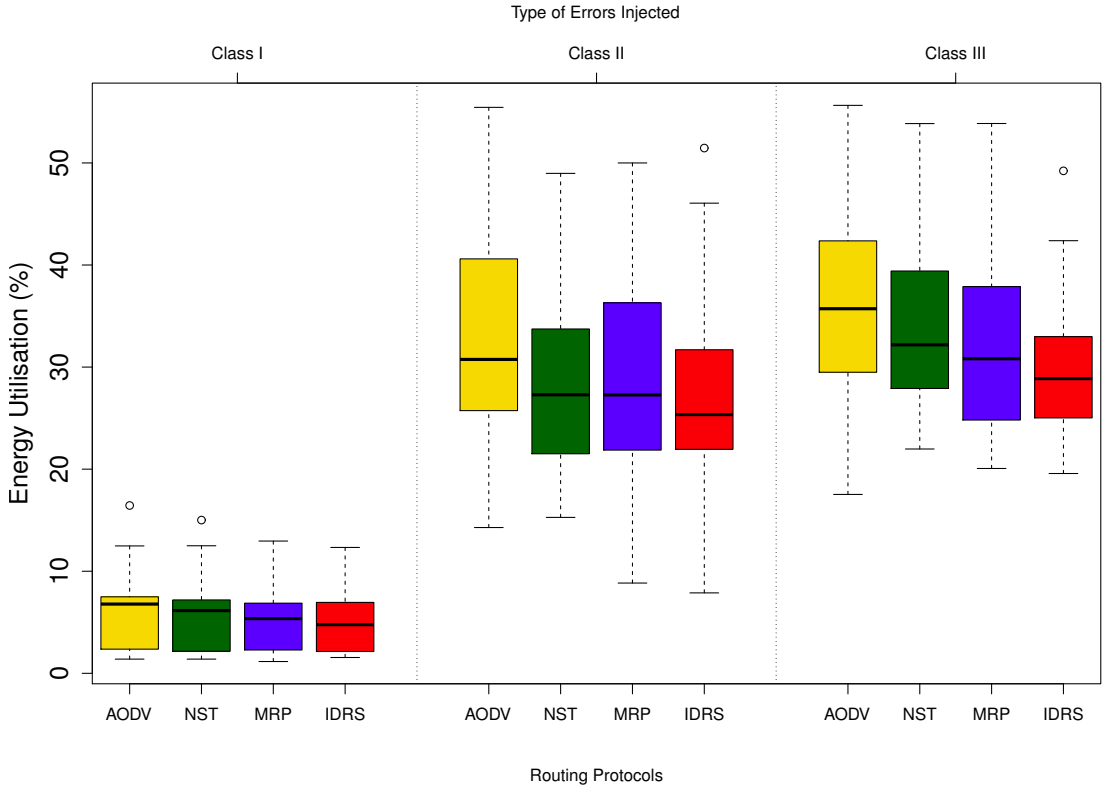



(b) Simulated routing overhead for large indoor deployment

Figure 5.15: The Box Whisker plot showing the routing overhead produced by the AODV, NST, MRP and IDRS for both small controlled 5.15(a) and large indoor 5.15(b) environments induced with 3 different interference classes.

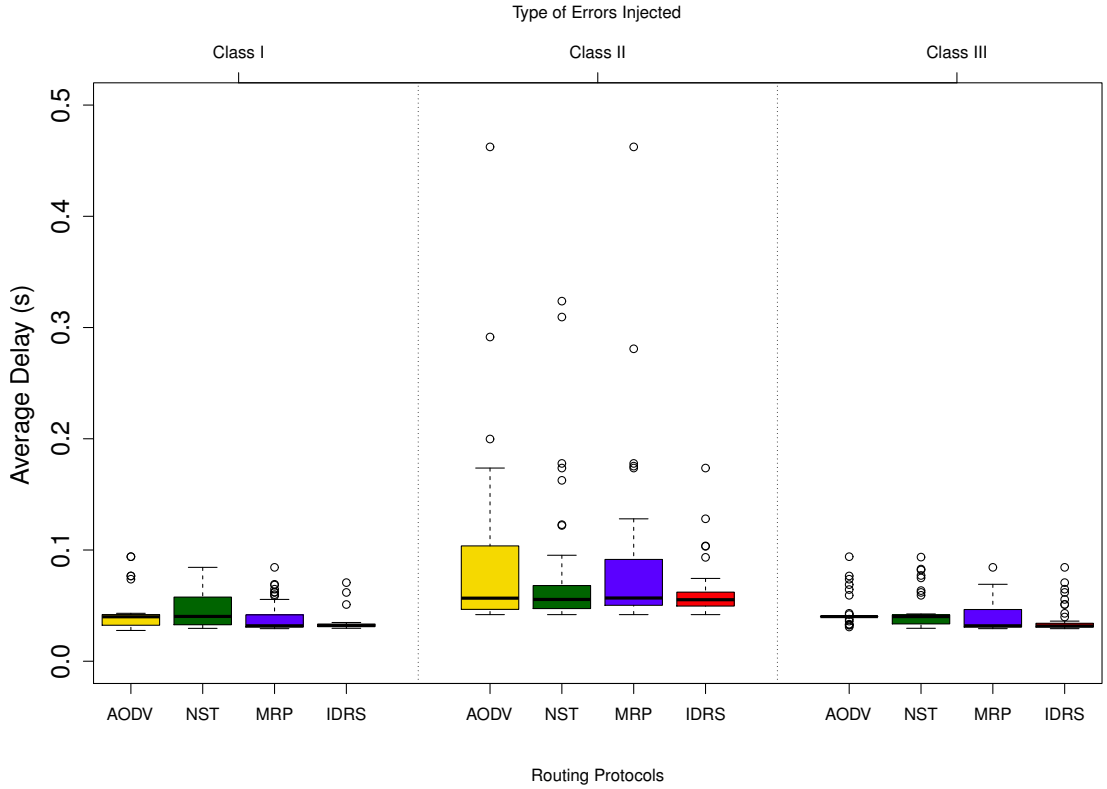



(a) Simulated energy utilisation for small controlled environment

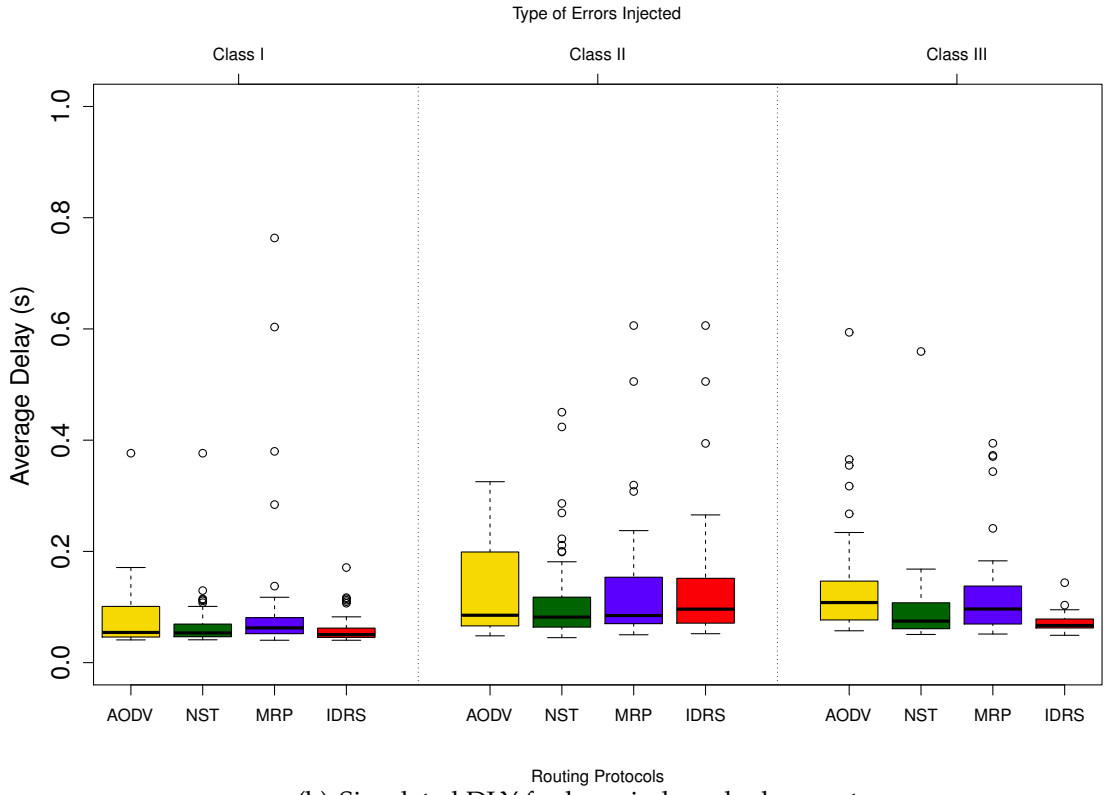



(b) Simulated energy utilisation for large indoor deployment

Figure 5.16: The Box Whisker plot showing the percentage of energy consumed by the AODV, NST, MRP and IDRS for both small controlled 5.16(a) and large indoor 5.16(b) environments induced with 3 different interference classes.




(a) Simulated DLY for small controlled environment




(b) Simulated DLY for large indoor deployment

Figure 5.17: The Box Whisker plot showing the average delay for the AODV, NST, MRP and IDRS during interference in both small controlled 5.17(a) and large indoor 5.17(b) environments.

Table 5.7: Significance test for AODV, NST and MRP against IDRS for the performance metrics PDR, ENG, RT and DLY in both small controlled and large indoor deployments. (Bold indicates 95% significance with $p$-value $< 0.05$ test and $A$-value $> 0.73$ or $< 0.27$ indicate large effect size.)

| ***Interference*** | ***Protocols*** | *PDR* | *ENG* | *RT* | *DLY* |
|---|---|---|---|---|---|
| ***Class*** | ***Controlled*** | | ***p-value*** | | |
| | AODV/IDRS | **1.16E-014** | **2.27E-010** | **1.60E-010** | **4.73E-007** |
| 3 | NST/IDRS | **0.00011** | **2.52E-011** | **6.93E-009** | **0.00005** |
| | MRP/IDRS | 0.44260 | 0.38965 | **0.01303** | 0.52409 |
| | AODV/IDRS | **1.03E-010** | **2.85E-008** | **0.00022** | 0.33348 |
| 2 | NST/IDRS | **0.00046** | **0.00085** | **0.01533** | 0.56283 |
| | MRP/IDRS | **0.03661** | 0.35647 | 0.09556 | 0.18960 |
| | AODV/IDRS | **2.25E-012** | **5.60E-009** | **6.14E-010** | **0.00229** |
| 1 | NST/IDRS | **0.02333** | **0.00080** | **0.00046** | **6.10E-005** |
| | MRP/IDRS | 0.80845 | 0.48434 | 0.41850 | 0.91627 |
| ***Class*** | ***Controlled*** | | ***A-value*** | | |
| | AODV/IDRS | **0.01677** | **0.10254** | **0.90070** | **0.81575** |
| 3 | NST/IDRS | **0.26647** | **0.09763** | **0.84926** | **0.74397** |
| | MRP/IDRS | 0.45036 | 0.44438 | *0.66000* | 0.54127 |
| | AODV/IDRS | **0.08424** | **0.14100** | **0.73913** | 0.56280 |
| 2 | NST/IDRS | *0.30568* | *0.31433* | 0.65262 | 0.54437 |
| | MRP/IDRS | 0.39933 | 0.46843 | 0.62371 | 0.60248 |
| | AODV/IDRS | **0.01435** | **0.09649** | **0.92823** | *0.71132* |
| 1 | NST/IDRS | 0.35610 | *0.28723* | *0.72228* | **0.75420** |
| | MRP/IDRS | 0.48421 | 0.45497 | 0.55205 | 0.50702 |
| ***Class*** | ***Indoor*** | | ***p-Value*** | | |
| | AODV/IDRS | **0.00020** | **0.00289** | **5.08E-011**1 | **1.12E-005** |
| 3 | NST/IDRS | 0.69946 | **0.01819** | **0.00795** | 0.07263 |
| | MRP/IDRS | 0.78175 | 0.27509 | **0.00641** | **0.00013** |
| | AODV/IDRS | **0.00110** | **0.00298** | **0.02827** | 0.85274 |
| 2 | NST/IDRS | 0.06638 | 0.30164 | 0.28315 | 0.29045 |
| | MRP/IDRS | 0.10740 | 0.24193 | 0.51853 | 0.82123 |
| | AODV/IDRS | 0.12441 | 0.08768 | **0.03204** | 0.27316 |
| 1 | NST/IDRS | 0.93739 | 0.64211 | 0.90832 | 0.25942 |
| | MRP/IDRS | 0.97907 | 0.74046 | 0.91700 | **0.01045** |
| ***Class*** | ***Indoor*** | | ***A-value*** | | |
| | AODV/IDRS | **0.25466** | *0.30129* | **0.90387** | **0.78623** |
| 3 | NST/IDRS | 0.47339 | *0.33999* | *0.67923* | 0.62232 |
| | MRP/IDRS | 0.51774 | 0.43071 | *0.67073* | **0.73614** |
| | AODV/IDRS | *0.30121* | *0.32077* | 0.63333 | 0.48841 |
| 2 | NST/IDRS | 0.38741 | 0.43654 | 0.56593 | 0.43506 |
| | MRP/IDRS | 0.40123 | 0.42815 | 0.53975 | 0.48593 |
| | AODV/IDRS | 0.40903 | 0.39861 | 0.62723 | 0.56513 |
| 1 | NST/IDRS | 0.49504 | 0.47163 | 0.51820 | 0.57849 |
| | MRP/IDRS | 0.49823 | 0.48005 | 0.50643 | 0.65270 |

environment, the reliability and availability of the MRP can be improved significantly even when the network size is increased from 6 to 51 nodes. From our results, the IDRS is scalable.

### 5.4.3 Discussion

Although the results from both the simulation and hardware experiments are not the same, the results from hardware experiments have shown that the proposed IDRS has improved the PDR and reduced the number of transmission. With the ability to determine the type of interference using the RDA, the appropriate response can be taken to rectify the failure. To evaluate the scalability of the IDRS, a large network of 51 nodes has been simulated using NS-2.34. The p-value and A-value computed from the simulated results have shown that the performance observed in IDRS is not always significantly better than AODV, NST and MRP when the network scales. However, the maximum energy utilisation and routing overhead generated by IDRS are lower as shown in the boxplot. As a future work, further investigation is required to analyse and tune the RDA to improve the detection and identification of the failures. The validity of the NS-2.34 simulator used to model the interference and evaluate the IDRS needs to be verified.

The plots in Figure 5.12 (PDR for Hardware) and 5.14(a) (PDR for Simulation) also demonstrated that the shape of the graphs are not the same. Although the simulation has been designed based on the hardware scenario and the failures are generated using the RSSI patterns collected from the motes, the results between hardware and software observed are not the same. We believe the dissimilarity may be caused by the error introduced the experiment. A systematic evaluation may be required to improve to improve the confidence between the hardware and software results.

## 5.5 Summary

In this chapter, we have presented an immune-inspired detection and recovery system called the IDRS. In the RDM, we have extended the RDA to diagnose different types of interferences. Our hardware experimental results have demonstrated that RDA can effectively classify the interference based on the RSSI values. Working together with the MRP, an effective response to network anomalies due to interference can be executed. The results in hardware sensor motes show that the IDRS can improve the PDR with Class I and III interference without generating excessive communication overhead. It can detect anomalies that affecting

the motes radio. As the signature of normal RSSI can be easily regenerated as required, the IDRS can be made to adapt to its changing environment. However, the hardware results are collected using a small network topology that may not model the real world and the IDRS results obtained from trace-based simulation have shown insignificant improvement on the performance. Additional works are required to validate the trace-based features implemented in NS-2.34 as a future work. Both the hardware and software also did not exhibit the same distribution. In order to reduce the gap between the hardware and software results, further investigations on the ability of the simulation to represent the real world is required.

CHAPTER

# SIX

# Using Statistical Approach to Demonstrate Dependability

In this chapter we are concerned with results that are generated via both simulator and experimental measurements. The results generated from multiple measurements can be subject to error. It is necessary to reduce the error by applying statistical analysis to summarise those observations and quantifies the uncertainty in the measured variable. We present the Systematic Protocol Evaluation Techniques (SPET), a comprehensive statistical approach for performing and evaluating the performance of a routing protocol to a level of confidence required. It utilises a combination of hardware and simulator, where the hardware acts a protocol conformation tool to ensure that the results are close to the real world deployments and the simulator allows us to perform extensive testing on the protocol.

The motivation for the work is introduced in Section 6.1. To improve the confidence of the results observed, the SPET is proposed in Section 6.2. A case study is presented in Section 6.3 to evaluate the benefit of SPET against the current state of the art approach followed by a discussion on the results from the case study. This chapter ends with a summary in Section 6.4.

## 6.1 Motivation

In Chapter 3, we have formulated a systematic evaluation methodology to evaluate the routing protocols proposed in this thesis. Existing state of the art techniques including the simulation and real hardware deployments have been applied to assess the dependability of the protocol. However, results from Chapter 5

have shown dissimilarity between the results obtained from simulator and hardware. These experimental techniques applied are susceptible to uncertainty that may make the outcome invalid and undependable. As a result, there is a need to reduce the uncertainties. There are two types of uncertainty namely: aleatory uncertainty and epistemic uncertainty (Helton, 1997). Epistemic uncertainty arises from the incompleteness and lack of knowledge of the system components of the sensor network where aleatory uncertainty occurs due to the random behaviour of the networks which changes with respect to time and its environment. Aleatory uncertainty may not produce the same result even with the same experiments calling for a more systematic approach to reduce this uncertainty. The issue of establishing confidence and minimising uncertainty in the results produced by these experiments has been widely ignored and the use of scientific empirical approach to reduce epistemic uncertainty is limited in WSNs. This objective of the chapter is to investigate the application of statistical techniques to reduce these uncertainties and provide the confidence of the results to address the research question:

> *Can the dependability of network protocols be demonstrated by reducing the experimental uncertainty using state of the art statistical techniques?*

## 6.2 Scientific Protocol Evaluation Technique

It can be very difficult to validate the reliability of the WSN's routing protocol even with repeated simulation because of the stochastic nature of the simulator. With simulation, more testing can be performed but the evidence from simulation is not always sufficiently realistic to be classed as valid (Zhao, 2005; Wu et al., 2010b). Even with the availability of real hardware to verify the simulator, it can be difficult to perform sufficient tests on real hardware to collect sufficient data in order to minimise the uncertainty and achieve the confidence required to show that a routing protocol can achieve the desired level of reliability. In order to overcome these problems, we develop a scientific approach, using statistical tools, to evaluate the routing protocol reliability in WSNs and cross validate the results acquired from the simulation against hardware experiments. The approach is known as SPET. Figure 6.1 shows the stage of the evaluation process, where each stage will be elaborated in the following subsection.

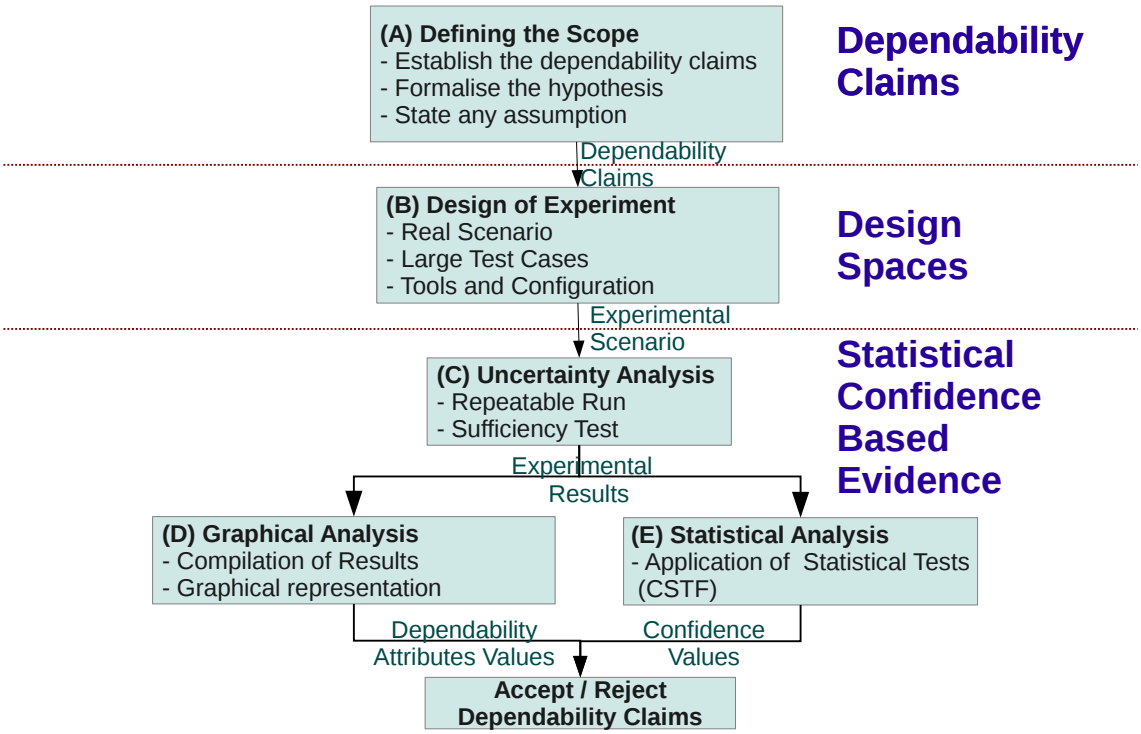



Figure 6.1: SPET consists of three parts: (I) define a set of dependability claims based on a specific problem domain. (II) specifies the design space of the experiment and the sample size required. (III) provides the evidence to support the dependability claims.

### 6.2.1 Defining the Scope

Before any experiment is carried out, we need to determine what we are trying to show by conducting this experiment. The reliability threshold for dependability needs to be defined according to the application requirements and the experimental boundary needs to be set to allow us to focus on the problem we are trying to solve. This can be done by stating the dependability claims that form the objective of the experiment. This will allow us to formulate the hypotheses focusing on the specific question to be answer. A hypothesis test can then be applied to determine if new data confirms or rejects the null hypothesis. The null hypothesis usually assumes no performance differences between the two samples. The alternate hypothesis assumes there is a change in the value between the two samples. By applying the correct statistical test, the outcome of the hypothesis test will allow us to either confirm or reject the null hypothesis. Any assumption made with regard to the experiment need to be highlighted. This step will demonstrate the empirical validity of dependency and the correctness of our approach.

### 6.2.2 Design of Experiment

This is an important phase to ensure the experiment is *credible* and *empirically valid*. There is a need to obtain realistic data that represent the true operational envelop of the system, to be used in the statistical test, to ensure the validity of the test. The accuracy of the data collected highly depends on the experimental approach used, and details of the experimental design taken and setup have to be documented and made available for download. Only with this information can we show that the experiments have been carried out properly with minimal error and unbiased (Kurkowski et al., 2005). Such information also allows other researchers to repeat the experiments and generate the similar set of data for comparison.

### 6.2.3 Uncertainty Analysis

In order to minimise the aleatory uncertainty in an experiment, it is necessary to acquire sufficient number of results by repeating the experiments $n$ times. This is usually not shown or performed in existing WSNs experiments as it can be a difficult and time consuming task to perform. $n$ can only be determined by repeating the experiment until a desired level of confidence is achieved. In SPET, this level of confidence is achieved statistically by comparing the effect size between first $n$ result samples against the next $n$ results samples acquired from the experiment using the same set of parameters. This process in SPET is known as sufficiency test. To perform the comparison, we apply the approach proposed by Read et al. (2011) and use the Vargha-Delaney $A$-test (Vargha and Delaney, 2000) to measure the effect size (differences) between the two samples. $A$-test gives an $A$-value in the range $[0, 1]$ to measure the differences. The $A$-test result will exhibit a no effect size ($A$=0.5) when a sufficient number of result samples $n$ is obtained.

### 6.2.4 Graphical Analysis

At this phase of the experiment, the simulation results are interpreted and presented. To ensure that the protocols are performing as expected, rigorous performance analysis using statistical methods and interpretation of the results are required. It is common in many works to report the mean of a number of runs, and in some cases with 95% CI error bar. The benefit of using the CI error bar is that we can determine whether the difference between the mean values is significant when the CI error bar do not overlap. However, when the CI overlap, the means may or may not be statistical significance. In order to assess statistical

significance, the sample size as well as variability has to be taken into account. Therefore, by analysing the error bars does not always allow us to determine whether the difference is statistically significant. The use of such statistics also relies on an underlying assumption that the results follow a normal distribution. Therefore, a safer alternative is to assume results do not have a normal distribution and employ non-parametric statistics (Helton, 2008).

Non-parametric techniques are suitable for use on data that is normal and non-normally distributed, and therefore in many cases more appropriate. Graphical representations such as scatter plots and box plots can identify unusual observations in the data that may be considered as outliers. These outliers need to be identified and analysed to minimise the likelihood of statistical error. In SPET, the box-whisker plot is used to analyse the results. The boxplot is suitable to analyse the uncertainties instead of error bar as critical information may be omitted during the calculation of means and deviation (Helton, 2008).

### 6.2.5 Statistical Analysis

In this section, a statistical approach is used to analyse the significance and confidence of the experimental results that will help to determine the validity of the dependability claims (hypotheses) formulated earlier and answer the following questions:

- What is the probability that the results are different?
- If it is different, is it significant?

Using the two statistical tools introduced in Section 3.1.6 of Chapter 3 namely statistical significance test and scientifically significance test, a Conceptual Statistical Test Framework (CSTF) in Figure 6.2 is proposed to analyse and verify the improvement observed are significant and have scientific values using the results generated from hardware and simulator.

The processes of CSTF are described as follows where Step 1, 2, 3 correspond to the labels (1, 2, 3) in the Figure 6.2.

**Step 1: Statistical Significance Tests**: *To determine whether there is any differences in the results produced from hardware and software.*

1. Presents the results from two sets of experiment from the hardware and simulation to a non-parametric statistical test (Wilcoxon test) to compute the $p$-value.

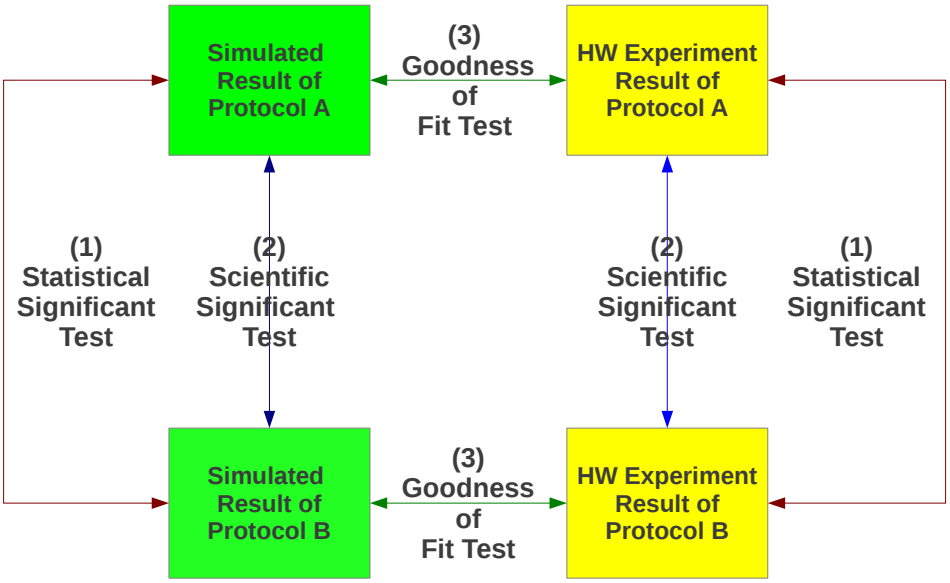



Figure 6.2: Conceptual Statistical Test Framework applies non-parametric tests to compare the distributions of the results obtained from the experiments and simulations

2. Apply the null hypothesis $H_0(1)$ to perform the hypothesis test. The $H_0(1)$ states that there is no difference between the two results.
3. Using a pre-determined confidence level of X%, $H_0(1)$ can be rejected if the computed $p$-value $<= \alpha$, indicating that our results are statistically significant. An $\alpha$ value of 0.05 is typically used, corresponding to a 95% confidence levels (Wilcoxon, 1945).

**Step 2: Scientific Significance Tests**: *To determine the degree of differences in the results generated from hardware as well as from simulation.*

1. Repeat Step 1 by using the same set of results to a non-parametric scientific significance test (Vargha-Delaney test) to determine the $A$-value.
2. Using the test statistic calculated from previous Wilcoxon test in Step 1, compute the $A$-value to measure the effect size between the two samples. The effect size is a measure of the strength of a phenomenon or the degree of differences (Vargha and Delaney, 2000).
3. If the $A$-value is $< 0.27$ or $> 0.73$, the difference between the two samples has a large effect size (large different) and is scientifically significant.

**Step 3: Goodness of Fitness Tests**: *To determine whether the simulated results is a*

*good representation of the real hardware.*

1. Take two sets of results, one each from hardware and software, and apply a goodness of fitness test (Kolmogorov-Smirnov (KS) test) to measure their discrepancy and to deduce the similarity between the results. The KS test is a non-parametric test used to determine whether two samples are drawn from the same distribution, by quantifying the distance between the empirical distribution functions of two samples.
2. Compute the *p*-value.
3. Apply the null hypothesis $H_0(3)$ for the KS test to accept or reject the hypothesis. The $H_0(3)$ states that the samples are drawn from the same distribution and we can reject $H_0(3)$ if the *p*-value $<= 0.05$ at 95% confidence levels.

### 6.2.6 Measuring the Effectiveness of the WSN's Evaluation

In order to demonstrate a dependability routing, we propose the use of the 3Cs approach commonly applied in requirement engineering to check the Correctness, Consistency, and Completeness (3Cs) of the evaluation approach proposed by Zowghi and Gervasi (2003). They have defined correctness as the ability to achieve completeness and consistency as well the ability to achieve the desired goal. In addition, we also consider the fourth C, Cost-effectiveness. For this section, we will define and discuss how these terms are used in this chapter:

- ***Correctness****: The ability to achieve consistency and completeness in our test results*

  The main challenge in designing experiments is to determine the total number of runs and consequently decrease the required time and cost while achieving its correctness. Zowghi and Gervasi (2003) define correctness as the satisfaction of achieving the specification of the test requirements. Statistical analysis allows us to inspect the data and extracts all the important details, taking into consideration the variability and measurement error that occurs during the experiment. It allows us to establish the confidence level of the test results and the correctness of the resultsin achieving the goals. It can help to prevent any unjustified claims about the effect of an algorithm. It is possible for an effect to be statistically significant, but of little or no scientific value (Vargha and Delaney, 2000). Hence, extensive statistical tests should be applied to the outputs in order to show the experiments are performed correctly and the magnitude of any significant effects must

be reported to indicate its precision with respect to its respective statistical values.

- ***Consistency****: The outcome of the tests does not contradicts each other under similar test conditions*
  Sensor networks can exhibit different behaviours across space, time and scale affecting the results acquired. The results obtained from an experiment may differ from another as it is exposed to the aleatory uncertainty. Aleatory uncertainty arises due to the stochastic nature of the system and can occur not only simulation (Mal-Sarkar et al., 2009), but also in real WSNs. This is mainly due to the pseudo-random number generator in the simulator or the non-deterministic behaviour of the sensor hardware, that can affect the output even if the same set of input parameters and experiment configurations are applied (Pawlikowski et al., 2002). Hence, it is necessary to perform uncertainty analysis by repeating the experiment sufficiently in both hardware and simulation to reduce the impact of aleatory uncertainty and to achieve consistency and sufficient level of confidence that is a representative of the prognostic results. By repeating the experiment allows us to determine and show consistency in the result.

- ***Completeness****: The test entails of everything that is required for a test condition to be true and result provides every aspect of the possible outcomes*
  A good experiment needs to be performed rigorously and in an unbiased manner to ensure its validity especially when performed in simulation. The experimental constraints and controls, such as the network parameters and scenario, used in the simulation should be consistent and must be set according to the aspect of real WSNs being studied to reduce epistemic uncertainty (Di Martino et al., 2012). Information such as realistic environmental model and application characteristics must be collected from real hardware and applied to the simulation to exercise the protocol under investigation and reduce the impact of epistemic uncertainty. Any initialisation bias needs to be addressed, and the experimental input must be randomised and tested on different scenarios rigorously.

## 6.3 Case Study: Evaluating the dependability of MRP using SPET

In order to explore the benefits of SPET, a case study to investigate the dependability of MRP against AODV and NST is conducted in this section. In order to demonstrate that the MRP can achieve the required dependability and its performance is significantly better than AODV and NST, we apply the SPET to perform the comparison and analyse the results collected from the simulation and hardware experiment.

### 6.3.1 Defining the Objectives

In this section, we will evaluate the performance of MRP against AODV and NST. In order to evaluate the dependability of MRP, we define the three main objectives (Dependability claims) of the experiments as:

- **Objective E-1**: to demonstrate that MRP can achieve the desired network reliability (90%) for different test cases compare to NST and AODV.
- **Objective E-2**: to demonstrate that MRP does not consume more resources than AODV and NST and is more efficient to reach the reliability target.
- **Objective E-3**: to demonstrate that the simulation results obtained are a valid representation of the real WSN.

In order to perform the test to achieve these objectives, we formalise a set of hypotheses as:

**Null Hypothesis 1 (**$H_0(1)$**)**: There is no difference in the number of packets delivered by MRP, AODV and NST.

**Null Hypothesis 2 (**$H_0(2)$**)**: There is no difference in the number of routing packets generated by MRP, AODV and NST to deliver a packet.

**Null Hypothesis 3 (**$H_0(3)$**)**: There is no difference in the PDR and routing overhead (RO) between the simulation and the real hardware implementation.

### 6.3.2 Experimental Setup

#### TelosB Experimental Setup

In the hardware experiment, we have set up a network as shown in Figure 6.3 using six TelosB motes. We have developed a test application (included in the MRP source code[1] to collect traffic measurements require for the comparison of those routing protocols. Each node is placed in a position of 0.5 metres from the other node so that it can only communicate with its immediate neighbours. The transmission power in each node is also set to a range of about 0.75 metres to avoid any interference from other non-neighbouring nodes. Channel 26 is used to avoid any interference with other WLAN operating in the same band. An acknowledgement for each packet is enabled for link layer notification operation.

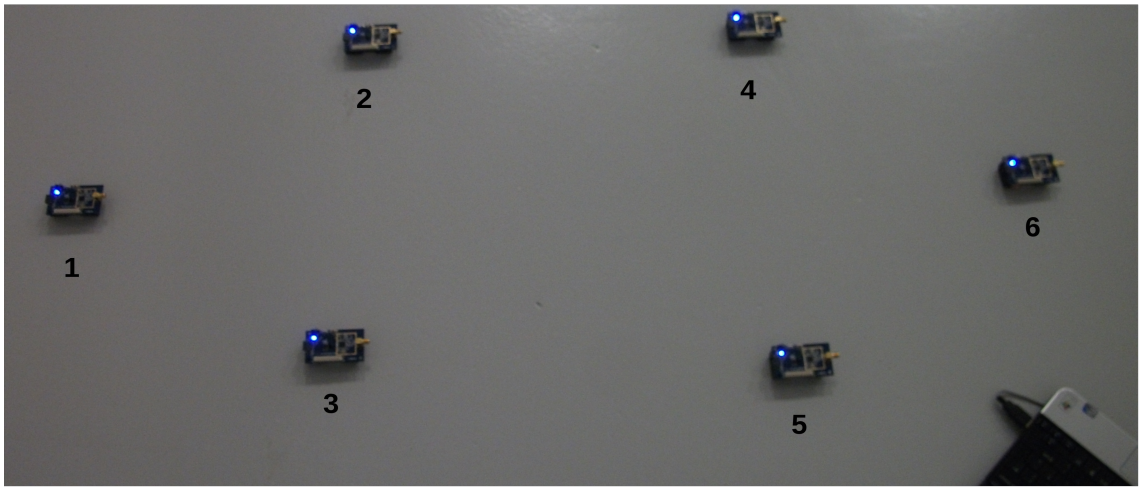



Figure 6.3: TelosB network setup using a small network for a better control of the experiment environment

In the experiment, Node 1 is configured to generate packet streams periodically and transmit the packet to base station (node 6) every 250ms via the intermediate nodes using the multihop routing protocol. Each experiment is run for 15 minutes (3600 packets generated) due to the storage limitation of the node to collect the statistics. Each packet includes the node id and a sequence number, which is incremented every transmission cycle and each packet is time-stamped. The network statistics, including number of packet sent, received, and the number of routing packet sent and received, are collected by individual node and stored in its on-board flash memory. These statistics are later retrieved from the individual nodes using a single hop communication by a node attached to a laptop for analysis.

[1]Downloadable at http://rtslab.wikispaces.com/file/view/mrptiny.tar

### NS-2.34 Simulation Setup

In order to validate the hardware against simulation, extensive simulations are performed on NS2. In Chapter 4, we have evaluated MRP and have achieved better performance on a larger network. For this simulation, we have designed a controlled experiment based on 6 static nodes that mirror the real world deployment in Figure 6.3. CBR traffic is used, in which node 1 is configured to transmit to node 6 at every 250ms (similar to the hardware experimental setup).

### Number of Runs

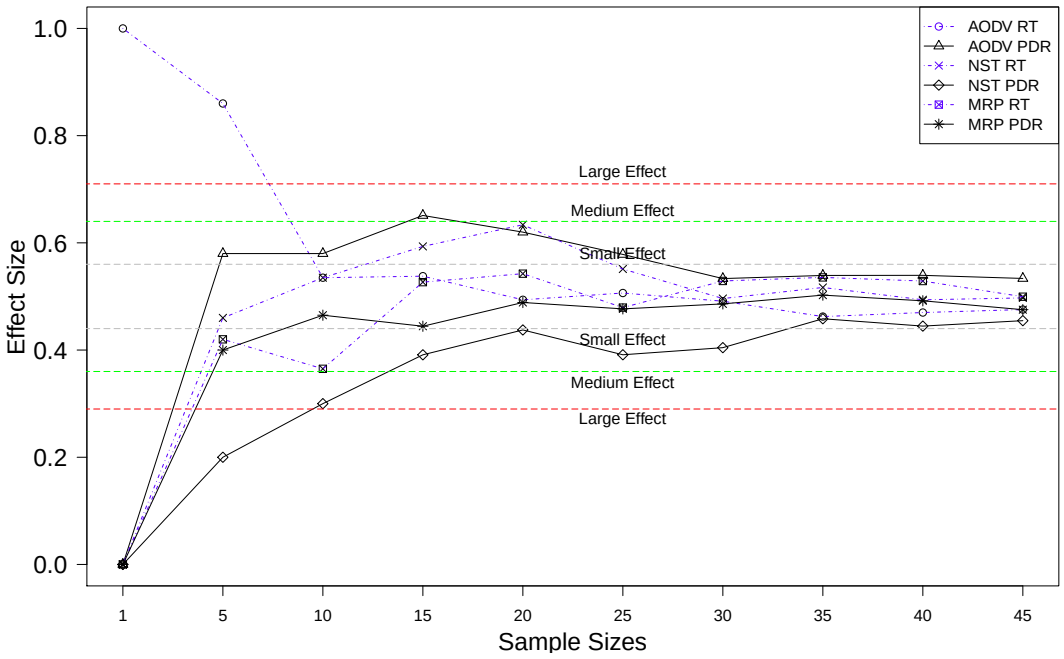
Due to the stochastic nature of the experiment, we perform the sufficiency test described in Section 6.2.3. Based on the technique proposed in Read et al. (2011), we have repeated the experiments $N$ times and compute the $A$-value until the $A$-value is $>0.45$ and $< 0.56$. This should ensure that we have collected enough samples for our experiments. From the results of the test shown in Figure 6.4, we have found out that the number of runs required for simulation is 35 while in hardware it is 15. Compare these numbers with the number of runs used by the experiments in Chapter 3, 4 and 5, all the repeated runs are above the minimum number of runs required. We have performed 50 runs in simulation and 15 runs in hardware.

### Test Scenario

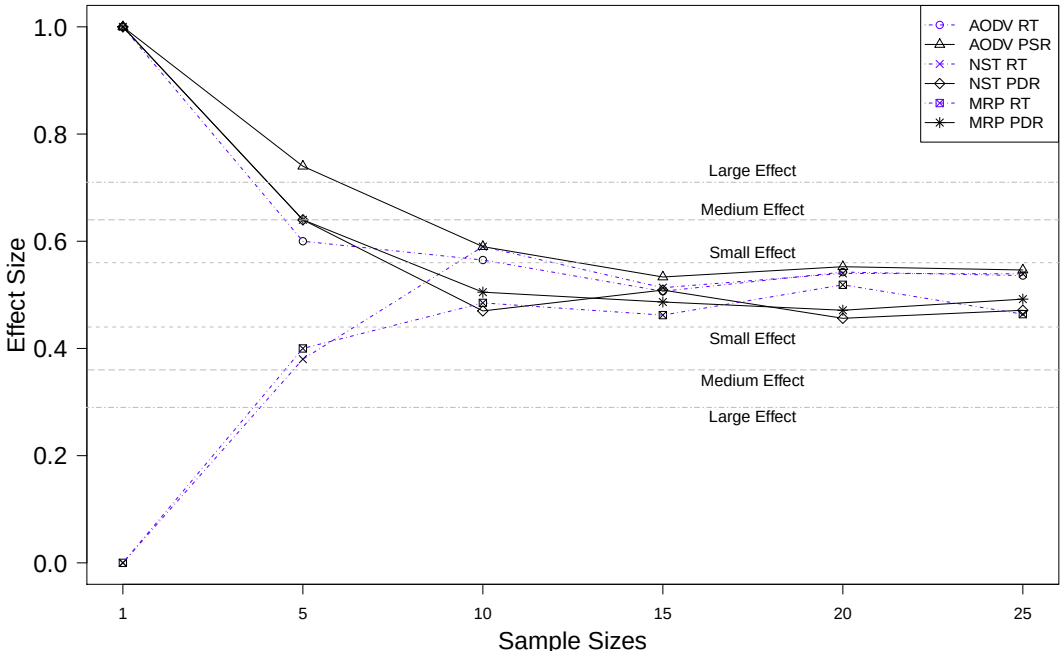
To investigate how irregular failure can affect the behaviour of the routing protocol and the overall network performance, transient failures are injected to the network based on arbitrary failure patterns with varying duration and frequency to mimic the real environments. We apply the ON/OFF approach presented in Chapter 3 by configuring an active node along the route not to respond during the MAC layer two-ways handshaking. These failures are only injected into the network after the initialisation period (10 seconds) required for the network to stabilise and to establish the initial route.

### Evaluation Metrics

Choosing the right metrics is crucial to assess the performance accurately. The two metrics proposed in Section 3.1.5 are selected as the input for the statistical tests to evaluate the statistical validity of the experiment. These metrics can be used to represent the reliability and efficiency of the protocol.




(a) Simulation: The $A$-value shows that a sample of at least 35 runs is required to bring the level of aleatory uncertainty within the small effect region




(b) Hardware: The $A$-value shows that a sample of at least 15 runs is required to bring the level of aleatory uncertainty within the small effect region

Figure 6.4: The $A$-values for different sample sizes (Number of repeated runs) from two set of experimental results: Simulation (a) and Hardware (b). In this test, the three effect size boundaries for the A test is indicated. We are interested within the small effect region where its represent there is no significant different between the samples.

- **Reliability:** PDR is used to measure the network reliability and is represented as the percentage of the number successful packet received $P_r$ to the total number of packet sent $P_s$. The higher the PDR, the more reliable the network is.

$$PDR = \frac{\sum P_r}{\sum P_s} \times 100$$

- **Efficiency:** The efficiency of the network, in term of the energy consumption, is usually represented by the number of routing packet required to transmit a data packet. It is calculated by normalising the sum of the total number of routing packets send to the total data packet received. A low value is desirable as it represents only a small amount of energy is wasted for communication.

$$Route_{overhead} = \frac{\sum Number\ of\ RREQ}{\sum Total\ Data\ Packet\ Received}$$

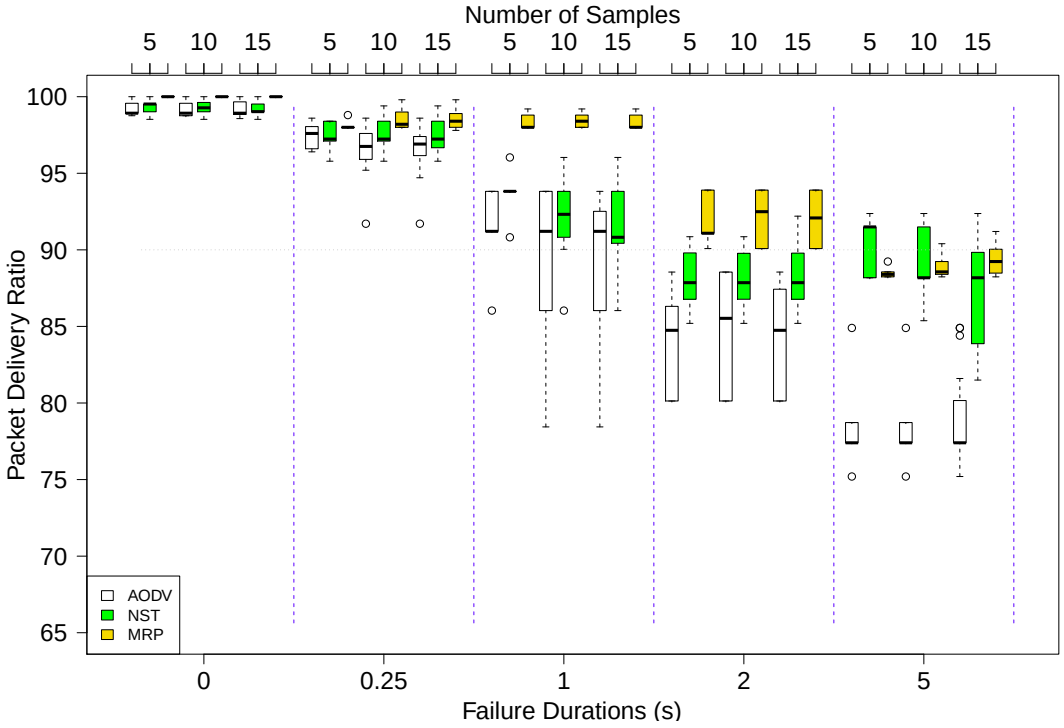
### 6.3.3 Results Analysis

The results obtained from both simulation and hardware are presented and analysed in this section.

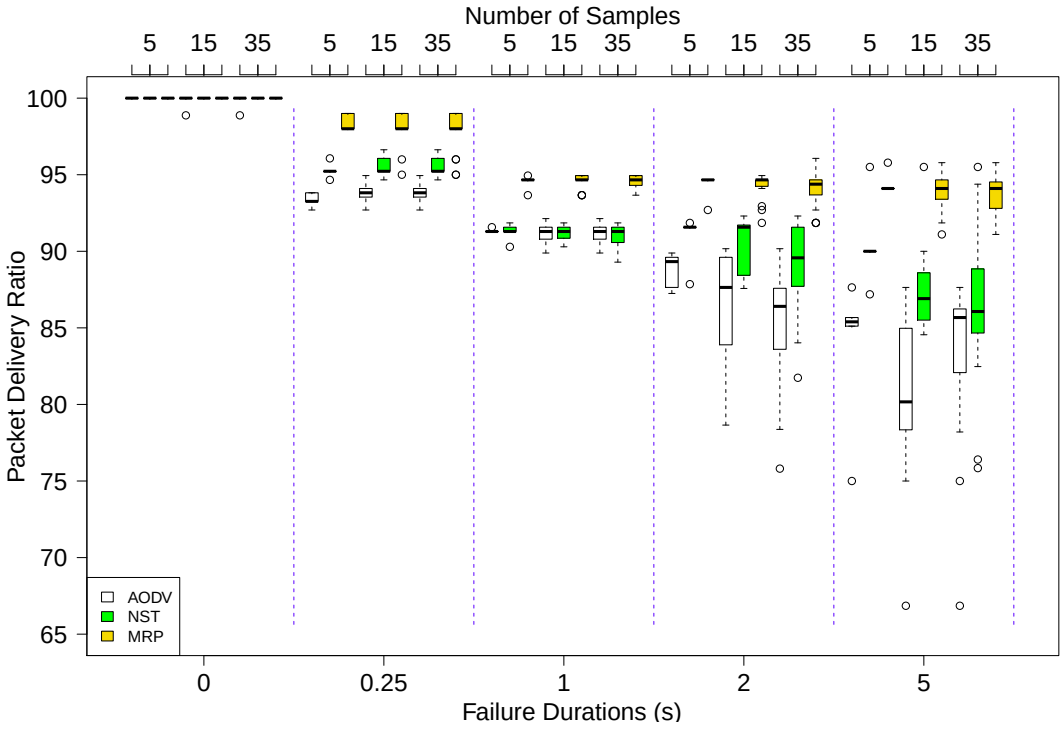
#### Reliability

One important performance metric used to measure reliability of a routing protocol is the success rate of packet delivery. Figure 6.5 and Figure 6.6 represent the median and the mean of the measured PDRs computed from the hardware and simulation results at five different failure durations: no failure, 0.25s, 1s, 2s, and 5s. In order to analyse the effect of the sample sizes on the computed results, 5, 10, 15 samples from hardware, and 5, 15, 35 from simulation are plotted.

- *Hardware Result:* By examining the boxplot, the performance variability between the three protocols can be easily understood compared to using the mean and error bar. We also observe the MRP has a higher reliability in delivering the packets than NST and AODV by looking at the median and IQR which is higher and narrow for MRP. MRP has achieved the 90% reliability target of the experiments. During normal network condition, about 3% data packets were dropped in AODV and 1% in NST due to random errors between the motes through collisions caused by initial RD. When the radios of node 4 and 5 were turned off at regular intervals (0.25s, 1s and

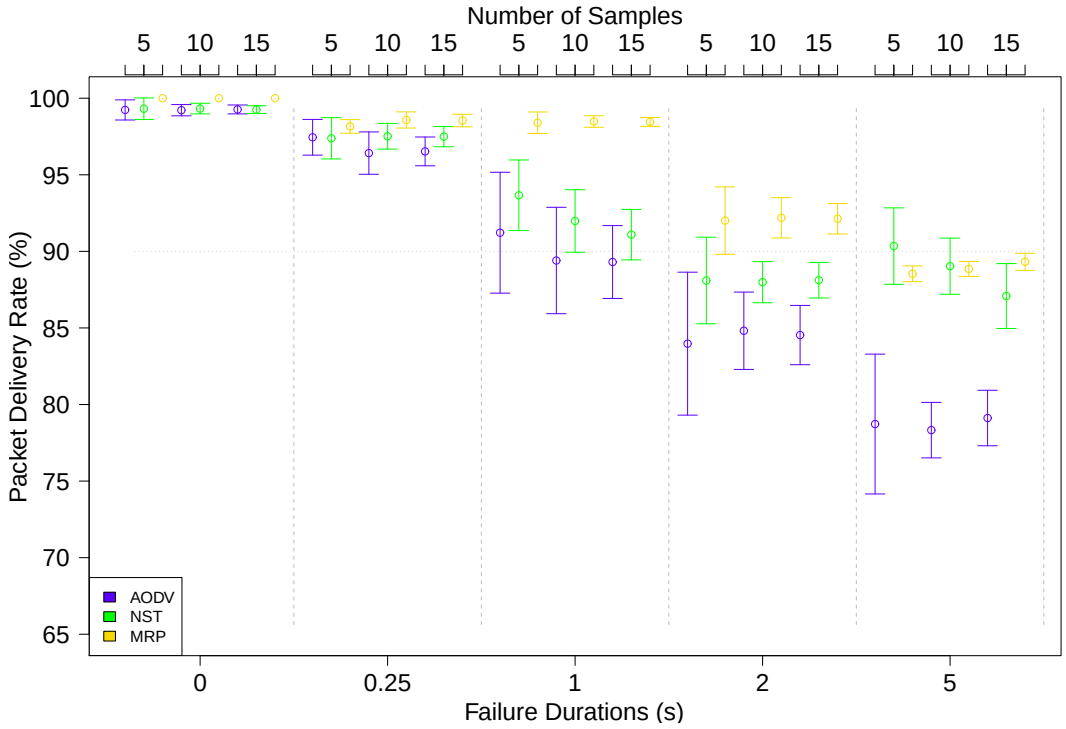



(a) PDR taken from 5, 10, 15 TelosB experimental runs

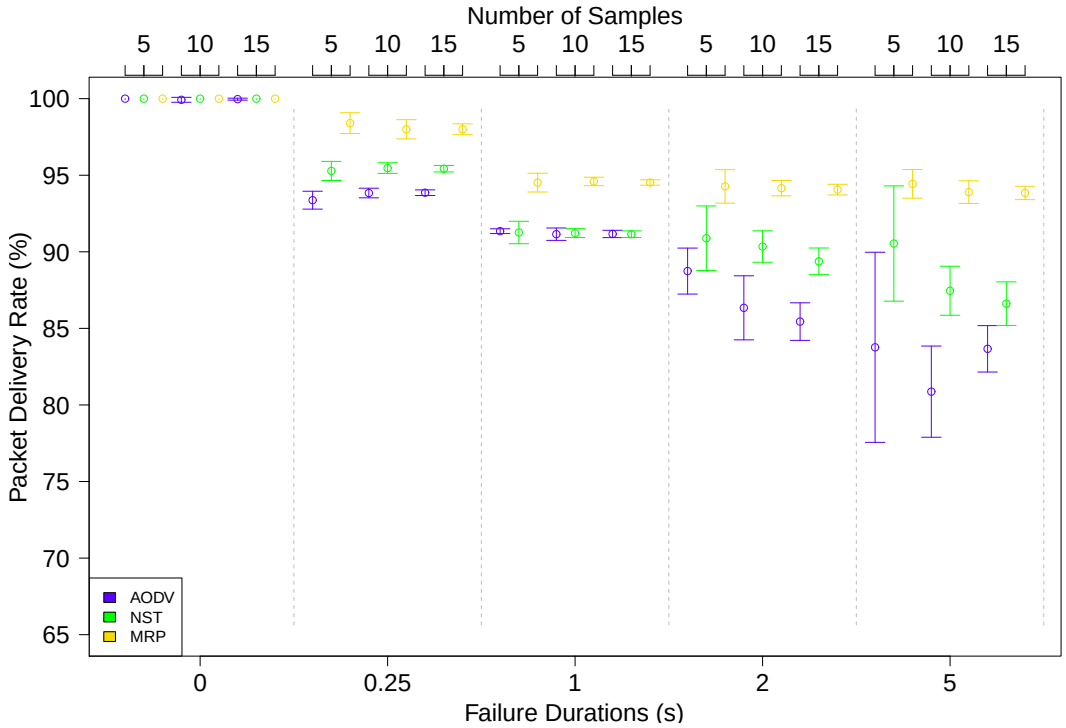



(b) PDR taken from 5, 15, 35 NS2 simulation runs

Figure 6.5: The Box Whisker plot showing Median (Black bar) , Inter-quartile range (Region inside the box), Max and Min values within the 1.5 IQR (Whisker) and Outliers (Dotted point) of the PDR for different sample sizes collected from TelosB hardware experiment (a) and from NS2 simulation. (b), with different failure durations.




(a) PDR taken from 5, 10, 15 TelosB experimental runs




(b) PDR taken from 5, 15, 35 NS2 simulation runs

Figure 6.6: The Mean-Error plot showing the Mean (Circle), and 95% CI (Error Bar) of the PDR for different sample sizes collected from TelosB hardware experiment (a) and from NS2 simulation (b), with different failure durations.

2s), MRP has achieved above 90% PDR compared to AODV and NST. The congestion caused by the routing packet in AODV and NST has prevented other nodes to transmit their data packets successfully. At 5s, the PDRs for MRP spread around the median of 90% and peaks at 92%. MRP has managed to achieve less than 10% packet loss compared to 20% in AODV, and 11% in NST as shown in Figure 6.5(a). Although we can see that the mean of MRP is higher than NST in most cases, we may not be able to determine whether there is a significant difference in performance between MRP and NST at 5s and 0.25s as they have overlapping CI. It is necessary to perform statistical additional tests to computer the $p$ and $A$-values . From the results of these tests, we can then deduce that the performance between the three routing protocol is significant with the $p$-value $< 0.05$ test and $A$-value $> 0.73$.

Table 6.1: $p$ and $A$ values for Hardware Experiment (Bold highlights significance value)

| Failure Duration | Routing Protocols | PDR | |
|---|---|---|---|
| | | $p$-value | $A$-value |
| Normal | MRP/AODV | **2.643347e-05** | **0.9** |
| | MRP/NST | **8.337457e-06** | **0.9333333** |
| | NST/AODV | 0.8347418 | 0.5244444 |
| 0.25 sec | MRP/AODV | **0.0002479539** | **0.8933333** |
| | MRP/NST | **0.03145654** | **0.7311111** |
| | NST/AODV | 0.1294403 | 0.6644444 |
| 1 | MRP/AODV | **2.432901e-06** | **1** |
| | MRP/NST | **2.432901e-06** | **1** |
| | NST/AODV1 | 0.4476234 | 0.5822222 |
| 2 sec | MRP/AODV | **2.52458e-06** | **1** |
| | MRP/NST | **3.39453e-05** | **0.9422222** |
| | NST/AODV | **0.005221419** | **0.8** |
| 5 sec | MRP/AODV | **2.996304e-06** | **1** |
| | MRP/NST | **0.03036426** | **0.7333333** |
| | NST/AODV | **4.204401e-05** | **0.9377778** |

- *Simulation Result:* Figure 6.5(b) shows the box-whisker plot of PDR obtained NS-2 simulations. During normal operation, when no fault was injected to the networks, all the three protocols have achieved 100% PDR. When faults were injected into the network, by turning off an active node along the two possible paths, 5% packet loss were observed in AODV due to the unavailability of the next hop neighbour. These numbers kept on increasing as we gradually increased the duration of failure in all the three

routing protocols. With the RSM in MRP, it have achieved above 90% packet received compare to 80% in AODV and 87% in NST. In terms of performance improvement, this is over 10% in AODV and 5% in NST at 5s. The same improvement can be observed from Figure 6.6(a). The non-overlapping error bar shows that they are significantly different at 95% CI. This is verified by the small $p$-values ($<<0.05$) in Table 6.2. A large effect size ($> 0.9$ and $<0.2$) is also observed for all failures. Hence, the reliability achieved by MRP is both statistically and scientifically significant.

Table 6.2: $p$ and $A$ values for Simulation (Bold highlights significance value)

| Failure Duration | Routing Protocols | PDR | |
|---|---|---|---|
| | | $p$-value | $A$-value |
| Normal | MRP/AODV | 0.3313349 | 0.5142857 |
| | MRP/NST | 0.3313349 | 0.5 |
| | NST/AODV | 0.3313349 | 0.4857143 |
| 0.25 sec | MRP/AODV | **2.459434e-13** | **1** |
| | MRP/NST | **1.302901e-10** | **0.9379592** |
| | NST/AODV | **4.358851e-12** | **0.02285714** |
| 1 | MRP/AODV | **3.717241e-13** | **1** |
| | MRP/NST | **3.916657e-13** | **1** |
| | NST/AODV1 | 0.9475107 | 0.495102 |
| 2 sec | MRP/AODV | **5.319198e-13** | **1** |
| | MRP/NST | **1.878333e-12** | **0.9877551** |
| | NST/AODV | **1.831988e-06** | **0.1681633** |
| 5 sec | MRP/AODV | **5.90433e-13** | **1** |
| | MRP/NST | **4.303105e-10** | **0.9334694** |
| | NST/AODV | **0.0496135** | 0.3632653 |

- *Comparing hardware against simulation:* We have also tried to validate the simulator against the hardware using KS goodness of fit test, on the samples collected from hardware and simulation. As highlighted in Table 6.3, only some test cases have shown similarity in PDR with $p$-value $> 0.05$. We believe this low number of similarity is caused by random radio noise in hardware that was not modelled in NS-2. During error-free condition, 1.2% of the total packets have failed to reach the destination in our hardware experiments in AODV and NST that was not observed in NS-2. The real sensor motes are sensitive to communication failures that can be tolerated by MRP.

- *Discussion:* From Figure 6.5 and 6.6, it is necessary to repeat the experiment and obtain sufficient samples to achieve an appropriate amount of

Table 6.3: KS Test $p$-values (Bold indicates two samples having the same distribution) for PDR between NS2 and TinyOS using Matlab

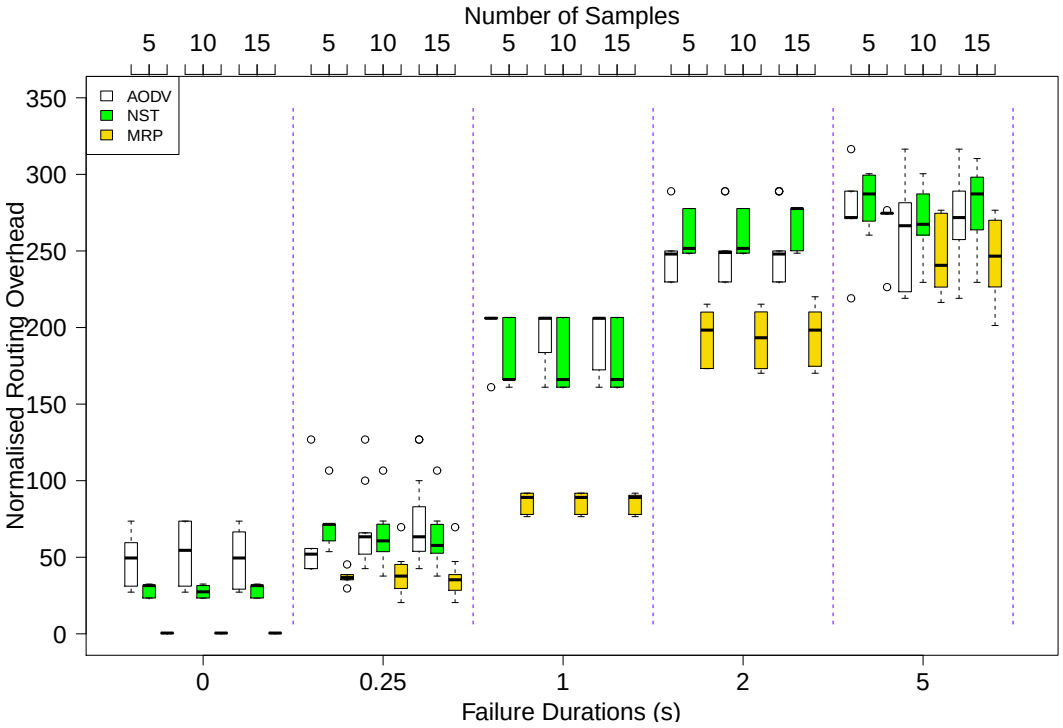
| $F_{rate}$ | Protocols | | |
|---|---|---|---|
| seconds | MRP | NST | AODV |
| 0 | **1** | 0.0001354575 | 0.0001354575 |
| 0.25 | **0.9250857** | 6.118046e-07 | 0.002545268 |
| 1 | 6.118046e-07 | **0.375211** | 0.07626242 |
| 2 | 0.0001354575 | **0.1813004** | 0.0006276553 |
| 5 | 4.229108e-06 | 0.009033161 | 2.558461e-05 |

variability in the result. This process is usually ignored in most papers that were reviewed. Using graphical approaches within our method (SPET), we have shown that MRP can achieve above 90% reliability at 95% confidence level with minimal number of test and unit time. The statistical analysis between MRP and NST, and MRP and AODV has shown that the difference in performance improvement is both scientifically and statistically significant. Hence, the null hypothesis $H_0(1)$ can be rejected. Although similar trends in reliability improvement are observed in both the hardware and simulation, results from the KS test have shown that the PDR between the hardware and simulation is statistically different.

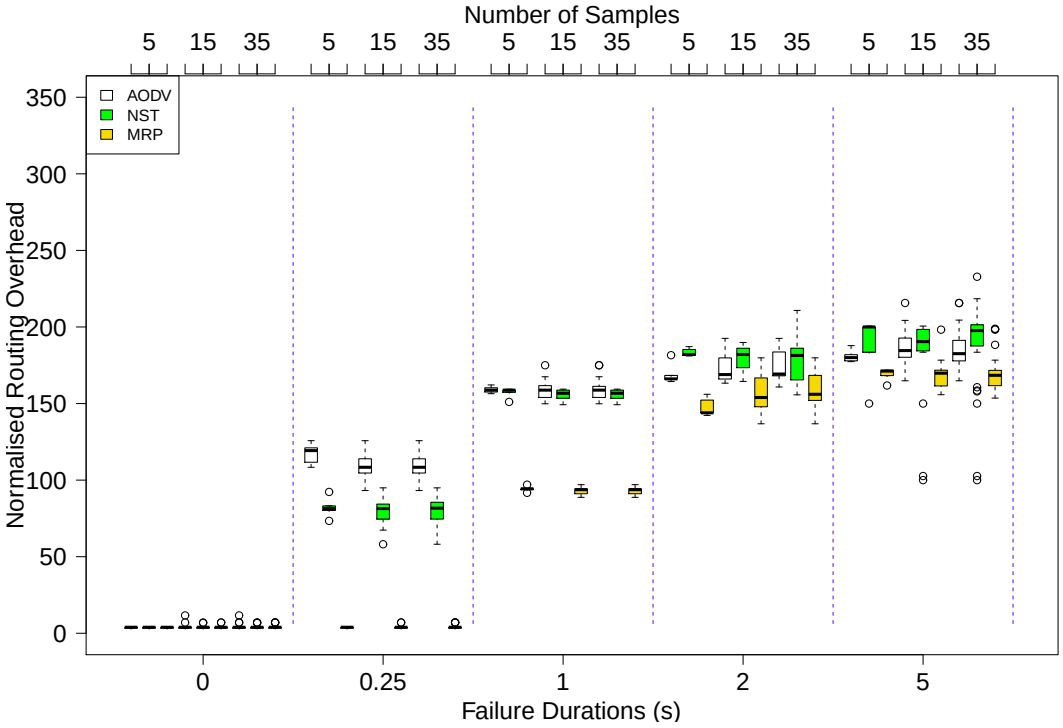
**Efficiency**

In WSNs, additional routing packets are required to establish a route to a destination when a link is unavailable. These additional packets are known to create additional overhead in the network and reduce the available of the network. It can cause the network to be unavailable for data packet transmission, and increase the energy consumption in the node due to additional transmission. In order to compare the energy efficiency of the routing protocol, we have analysed the RO between the three routing algorithms using box-whisker plot as shown in Figure 6.7 for different sample sizes with different failure duration. To compare the benefit of SPET, the mean and error plot for the RO is also provided in Figure 6.8.

- *Hardware Result:* When we increased the radio failure duration in the active node from 0.25 to 5 seconds, the RO also increased in AODV and NST. This overhead is significantly less in MRP between 0 to 2 second failure intervals as shown in Figure 6.7(a). The switching mechanism in MRP can abort RD and switch immediately to RT when the next hop neighbour is available for

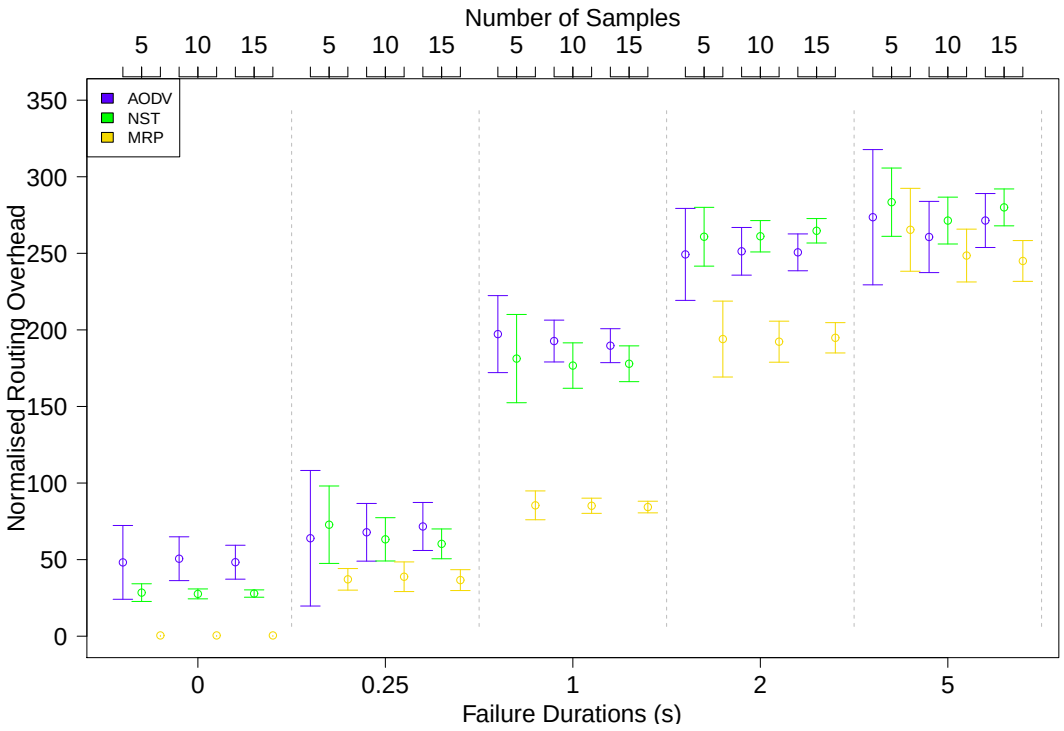



(a) Routing Overhead from TelosB Hardware

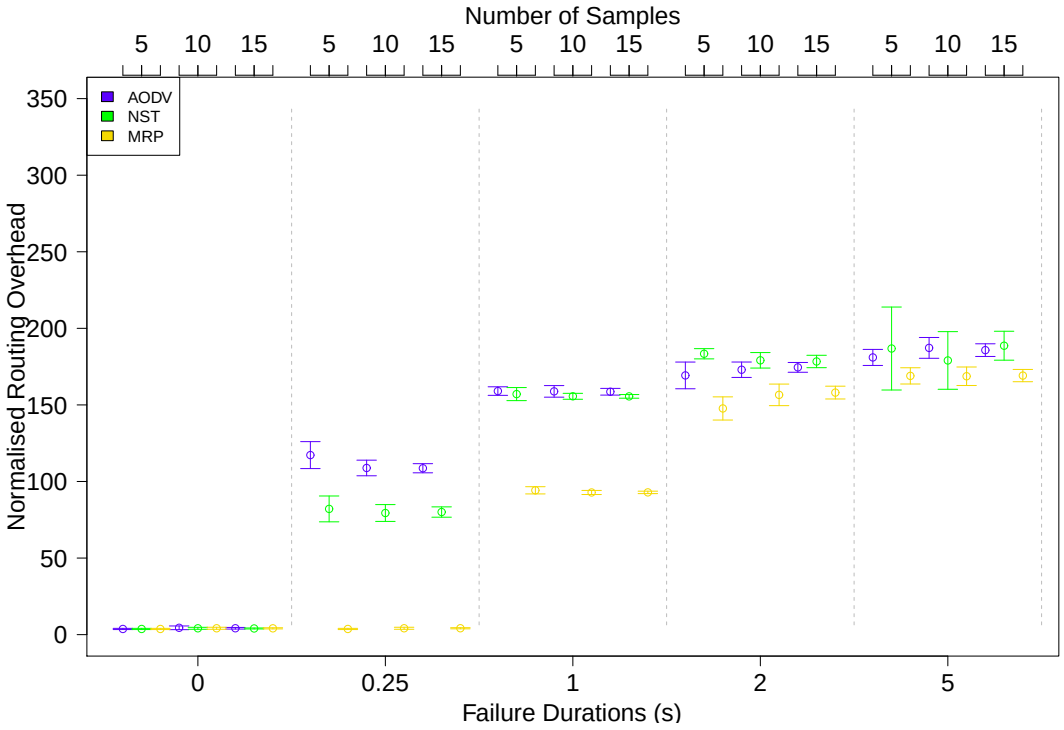



(b) Routing Overhead from NS2 Simulation

Figure 6.7: The Box Whisker plot showing the Median (Black bar) , Inter-quartile range (Region inside the box), Max and Min values within the 1.5 IQR (Whisker) and Outliers (Dotted point) of RO taken from different sample sizes collected from TelosB hardware experiment (a) and from NS2 simulation. (b), with different failure durations.




(a) Routing overhead from TelosB Hardware




(b) Routing overhead from NS2 Simulation

Figure 6.8: The Mean-Error plot showing the Mean (Circle), and 95% CI (Error Bar) of the RO for different sample sizes collected from TelosB hardware experiment (a) and from NS2 simulation (b), with different failure durations.

communication making it capable to operate more efficiently during transient random error. The RO for MRP gradually increases with failure duration until it reaches above 220 at 5 seconds, where this value is close to the RO of AODV. However, the difference between AODV and MRP cannot be observed with only 5 samples. Additional $n$ runs (where $n$=15) are required, that was determined statistically in SPET. From the mean error plot, the significant difference between AODV and MRP cannot be determined as the error bar in Figure 6.8 overlap at 5s for 15 samples. This can only be determined using statistical tests. Table 6.4 shows that the differences are scientifically significant as represented by the A-value in bold.

Table 6.4: $p$ and $A$ values for Hardware Experiment (Bold highlights signicance value)

| Failure Duration | Routing Protocols | Routing Overhead | |
|---|---|---|---|
| | | $p$-value | $A$-value |
| Normal | MRP/AODV | **6.264758e-07** | **0** |
| | MRP/NST | **76.326979e-07** | **0** |
| | NST/AODV | **0.01946062** | **0.2488889** |
| 0.25 sec | MRP/AODV | **7.840888e-05** | **0.07555556** |
| | MRP/NST | **9.558141e-05** | **0.08** |
| | NST/AODV | 0.318427 | 0.3911111 |
| 1 | MRP/AODV | **2.853591e-06** | **0** |
| | MRP/NST | **2.667462e-06** | **0** |
| | NST/AODV1 | 0.4668257 | 0.4222222 |
| 2 sec | MRP/AODV | **3.073756e-06** | **0** |
| | MRP/NST | **3.105197e-06** | **0** |
| | NST/AODV | **0.03296333** | 0.7288889 |
| 5 sec | MRP/AODV | **0.03417722** | **0.2711111** |
| | MRP/NST | **0.001204504** | **0.1511111** |
| | NST/AODV | 0.5067258 | 0.5733333 |

- *Simulation Result:* The simulated RO is shown in Figure 6.7(b). During normal condition, each successful packet received requires only 7 routing packets to be sent on average for all the routing protocols. This number increased linearly for NST and AODV with failure duration where they peaked at 2 seconds. However, the overhead in MRP is less than AODV and NST as shown by the mean and median in Figure 6.8(b) and is also statistically significant indicated, by a very small $p$-value ($<<0.05$) in Table 6.5. Hence, $H_0(2)$ can be rejected.

- *Comparing hardware against simulation:* By analysing the mean and median values in Figure 6.7 and 6.8, each failure scenario has shown differences in

Table 6.5: $p$ and $A$ values for Simulation (Bold highlights significance value)

| Failure Duration | Routing Protocols | Routing Overhead | |
|---|---|---|---|
| | | $p$-value | $A$-value |
| Normal | MRP/AODV | 0.8938737 | 0.5085714 |
| | MRP/NST | 0.8518442 | 0.5118367 |
| | NST/AODV | 0.9789139 | 0.5020408 |
| 0.25 sec | MRP/AODV | **2.850978e-13** | **0** |
| | MRP/NST | **2.841219e-13** | **0** |
| | NST/AODV | **1.056553e-12** | **0.995102** |
| 1 sec | MRP/AODV | **5.712717e-13** | **0** |
| | MRP/NST | **5.565661e-13** | **0** |
| | NST/AODV | **0.03690669** | 0.644898 |
| 2 sec | MRP/AODV | **1.221272e-06** | **0.162449** |
| | MRP/NST | **2.722869e-08** | **0.1134694** |
| | NST/AODV | 0.312294 | 0.4293878 |
| 5 sec | MRP/AODV | **3.422429e-07** | **0.1453061** |
| | MRP/NST | **1.309193e-05** | **0.1967347** |
| | NST/AODV | **0.008807238** | 0.317551 |

RO between the hardware and simulator. This is verified using the result computed by KS Test in Table 6.6 where all the failure scenarios have shown a small $p$-values ($<<0.05$). Hence, we cannot verify that the result produced by simulator is a valid representation of real hardware.

Table 6.6: KS Test $p$-values (Bold indicates two samples have the same distribution) for Routing Overhead between NS2 and TinyOS using Matlab

| $F_{rate}$ | Protocols | | |
|---|---|---|---|
| seconds | MRP | NST | AODV |
| 0 | 6.118046e-07 | 6.118046e-07 | 6.118046e-07 |
| 0.25 | 6.118046e-07 | 4.229108e-06 | 0.002545268 |
| 1 | 0.002545268 | 0.002545268 | 6.118046e-07 |
| 2 | 2.558461e-05 | 6.118046e-07 | 6.118046e-07 |
| 5 | 6.118046e-07 | 1.289345e-08 | 6.118046e-07 |

- *Discussion:* From Figure 6.7 and 6.8, the median and mean of RO in MRP is smaller than AODV and NST. This difference is both statistically and scientifically significant as shown by the low $p$-value and high effect value of $A$-value. Hence, the results reject $H_0(2)$. However, there is no evidence from the KS-Test to show that the overheads between the simulation and hardware are similar.

### 6.3.4 Discussion: Benefit of SPET

From our results, we have shown that SPET has improved the confidence of the results observed in a WSN's experiment by incorporating a comprehensive empirical approach and statistical techniques in the experiments. In order to show the proposed SPET meets the requirement to demonstrate dependability which is not covered by existing state-of-the-art (SOTA) approaches, we evaluate the properties of SPET based on the 4Cs requirement engineering approaches.

- **Better assessment of the significance:** To show the correctness and confidence level of the results, SPET has evaluated the results and validated the reliability achieved by one protocol (MRP) is both scientifically and statistically significant using extensive statistical tests as shown in Table 6.1, 6.2, 6.4 and 6.5. As for current SOTA approach applied in WSNs, there are two limitations: Firstly, the results are reported based on a 95% CI to support the hypothesis. CI is highly dependent on parameters used in the statistics such as the sample size. Hence, getting the right sample size is important. Secondly, hypothesis testing was never performed as the statistical significance of the results is always determined by looking at the overlapping of the CI error bar. However, employing hypothesis testing and appropriate analysis, we can deduce in Table 6.1 that all the PDR for MRP is statistically significant as the $p$-value $<$0.031 and it has large effect size (with $A$-value $>$0.73). Hence, $H_0(1)$, can be rejected. Hence, SPET allows us to demonstrate the correctness and provide the completeness of the results at 95% confidence level.

- **Better assessment of variability:** From our experiments, both the box-whisker and mean-error plots have shown that MRP has managed to achieve more the 90% PDR (Objective E-1) at a lower overhead (Objective E-2), which satisfies the dependability requirement defined in 6.3.1. However, the use of Box-whisker plot to visually analyse our results in Figure 6.5 allows us to understand the spread and variability of performance in more details such as the skewness and outlier, than mean-error plot. For example, Figure 6.6 allows us to deduce that the mean PDR for NST from hardware experiment (5s failure duration) only achieves 87.5%. However, the boxplot from Figure 6.5(a) allows us to further deduce that most of the PDRs observed below the average value and the difference between the maximum and minimum PDR is 10%. This information will allow one to determine the best and worst performance of the routing protocol which is not visible from mean-error plot. Hence, a complete characteristic of the result is obtained.

- **Better approach of dealing with sample size:** It is necessary to repeat the experiments to reduce the aleatory uncertainty in our experiments as insufficient sample size can affect the consistency of the result's statistics as shown in Table 6.7. A small sample size ($< 15$) may give incorrect median and means (in bold) while an excessive sample size does not provide additional information. The effect size between the two samples (in italic) is also affected by the sample size. However, determining the sample size can be time consuming and challenging. Existing evaluation uses the mean with 95% CI to analyse the result where it is necessary to repeat each experiment until a small CI is observed and there is no mention of how these number of runs is determined. Following our method, the sample size can be computed statistically using the results obtained from two similar experiments. With the use of the required sample size, we can ensure the consistency and correctness of the statistics computed while minimising the time required to perform the experiments

Table 6.7: This table presents the frequency (in $\%$) where one protocol is better than the other (bold indicate $< 95\%$ out of 50 runs) as the sample size increases. For a small sample, the probability for one protocol is better than the other is less compare to a larger sample. Too large sample may not provide useful information

| | NST>AODV | | MRP>NST | | MRP>AODV | |
|---|---|---|---|---|---|---|
| Sample Size | Median (Mean) (%) | High Effect (%) | Median (Mean) (%) | High Effect (%) | Median (Mean) (%) | High Effect (%) |
| 1 | **73.8** **(73.8)** | 100.0 (100.0) | **92.2** **(92.2)** | *98.0* *(98.0)* | 100 (100.0) | 100 (100.0) |
| 5 | **82.8** **(94.4)** | 67.4 (59.3) | 100.0 (100.0) | *96.0* *(96.0)* | 100 (100.0) | 100 (100.0) |
| 10 | **90.4** **(99.2)** | 62.4 (56.8) | 100.0 (100.0) | *99.2* *(99.2)* | 100 (100.0) | 100 (100.0) |
| 15 | **92.4** **(99.8)** | 60.6 (56.1) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |
| 20 | 95.6 (100.0) | 57.1 (54.6) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |
| 25 | 97.6 (100.0) | 55.3 (54.0) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |
| 30 | 99.8 (100.0) | 52.1 (52.0) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |
| 35 | 100.0 (100.0) | 58.2 (58.2) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |
| 40 | 100.0 (100.0) | 54.0 (54.0) | 100.0 (100.0) | 100.0 (100.0) | 100 (100.0) | 100 (100.0) |

- **Wider selection of test cases:** It is necessary to test the routing protocol with different test cases as incomplete tests may cause the routing protocol to fail during real deployment as well as affecting the quality of the results. This is usually defined with respect to the type of environment the networks are deployed in. However, it is not possible to implement and test all the real world phenomena that can affect the routing functionality in simulation. Hence, only test cases based on real WLAN environment has been applied. Although limited test cases were applied in our experiments, more complete test scenarios (if known) can still be applied in SPET to further reduce the epistemic uncertainty. SPET helps us to ensure the completeness of the test conducted.

- **Cost-benefit approach:** SPET provides a cost-effective and time-efficient approach for evaluating protocols. SPET applies a hybrid approach and the KS test to validate the performance of the routing protocol. Using this approach, it is not necessary to perform a large scale WSNs testing in both hardware and software in order to obtain sufficient confidence. We only need to deploy a small network and perform **N** number of runs in hardware to obtain enough results that will be verified against extensive simulations, that is scalable in size, in order to achieve the required confidence level. By doing so, the amount of time to perform the experiment will also be reduced.

  For instance, if

  $T_r =$ the time taken to run an experiment,
  $R =$ Number of random runs,
  $TC =$ Number of test cases,
  $P =$ Number of protocols to compare,

  then the total time requires to run the whole experiment:

$$T_e = N \times T_r \times R \times TC \tag{6.1}$$

  Therefore, in simulation, the total time taken $T_{sim}$=5250s where, P = 3, TC = 5, $T_r$ = 10s, and R = 35. In hardware, the actual time taken $T_{actual}$=202,500 seconds, where P = 3, TC = 5, $T_r$ = 900 seconds, R = 15. This implies that our experiments only took approximately 11 days to complete based on 5 hours work per day to carry out all the hardware experiments required. More experiments can be performed in simulation as it is faster to run. However,

if the same number of runs performed in simulation (R=35) is conducted in hardware, the estimated time will take $T_{predict}$=472500 seconds which is equivalent to 26.25 days, in which we have reduced the time taken to approximately half. Hence, with SPET, the experiments can be performed faster, cheaper and with confidence.

## 6.4 Summary

In this Chapter, we have presented a systematic approach to evaluate the results generated from the simulation and experiments called SPET. It uses non-parametric statistical tests to conduct and analysed the data collected in order to improve the confidence and reduce both the epistemic and aleatory uncertainties in an experiment. Our results have shown that SPET can reduce the experiment time and demonstrate the observation made is correct and consistent. Furthermore, as an alternative to the existing evaluation approaches, it is shown that not only the SPET does not show any unexpected effects on the results but it helps to improve and provide more confidence in the results to demonstrate the dependability of the routing protocol. As future work, the SPET can be extended to include a systematic experimental design technique to validate the model used in the experiments to further address epistemic uncertainty.

CHAPTER

# SEVEN

# Conclusions and Future Works

The hypothesis posed in this thesis is to investigate whether the multi-modal recovery and immune-inspired approaches can be applied to improve the network dependability of the WSNs, with a focus on the source of uncertainty arising from the failures that causes unpredictable degradation of the packets delivered. This is of interest and importance because current routing approaches are unable to cope with different failure conditions that occur in the dynamic wireless environments. Single routing approaches are usually designed and tested on specific network under controlled environments. This specific routing may yield incorrect responses that may worsen the degradation of the networks.

## 7.1 Significant Contributions of the studies

This section summarises the contributions in this thesis.

**A multi-modal approach toward network dependability**

In Chapter 4, a multi-modal concept is applied to the network layer of WSNs to improve the network dependability and ability to tolerate failure. From the dependability assessment, it was identified that the communication between two nodes are unreliable and prone to failure. The survey on the existing literatures conducted has revealed that unreliable communication can deteriorate the packet delivery reliability and the irregular failure occurring in the network can significantly impact the operation and performance of the network protocol. Simulated results presented in Chapter 3 have demonstrated that a network operating in a

single routing mode does not always improve the delivery rate and might not be able to tolerate nodes failures with different durations and density.

To address the problem, the MRP is proposed. The MRP integrates a number of routing protocols in the node. Each node is allowed to select and operate in a routing protocol that has a high success rate of forwarding the packet. Three reactive routing protocols are selected and implemented in the MRP to investigate the benefit of applying the multi-modal approach. The robustness and scalability of the MRP are evaluated in simulator. The simulated results have shown that the MRP can deliver more packets and utilise less resources than AODV and NST. The results also demonstrate that the MRP can tolerate the simulated faults better than AODV and NST. Hence, a multi-modal approach improves the network dependability of the WSNs

**Providing self-healing and assisted recovery in WSNs using AIS**

In addition to using the multi-modal approach to tolerate different failures, this thesis also investigates the application of the immune-inspired algorithms to provide fault tolerance and improve dependability. Based on a review on existing immune-inspired algorithms, it was found that the RDA has the ability to detect changes in a dynamic environment and can yield high positive rate to detect anomalies. It does not suffer from the curse of dimensionality or single point of failure. As a result, the RDA is extended to assist in identifying the characteristics of the interference detected.

To allow self-healing in WSNs, the IDRS is proposed in Chapter 5. IDRS mimics the immune systems multi-layer defence approaches where the MRP will provide the first line of defence to detect and perform recovery. If the fault is not rectified by the MRP, the MRP will activate the RDA to diagnose the failure and identify the fault. The interactions between the MRP and RDA will allow an appropriate response to be taken in order to rectify the failures.

Using the real TelosB hardware and NS-2 simulator, the performance of IDRS is compared against the MRP, AODV and NST. Interferences from a WLAN source are used to generate the failures between two communicating nodes. Results from both the experiments have showed that the IDRS have improved the packet reliability and the efficiency of the network. The dependability achieved by IDRS is significantly higher than MRP. The hardware results also show that the RDA can correctly identify interferences that can significantly degrade the PDR.

**Improving the confidence and reducing the reality gap using statistical approaches**

Different evaluation techniques such as simulation, testbeds and real hardware deployments have been discussed in Section 2.9 to assess the dependability of the protocol. Simulations can provide valuable information about a system's behaviour, but cannot replace the experiments of the system on real hardware. However, the use of statistical analysis in the results is limited in many of the published works. This may lead to false conclusions and affect the credibility of the works. These experimental techniques are usually susceptible to uncertainty that could make the outcome invalid. The issue of establishing confidence and minimising uncertainty in the results produced by these experiments has been widely ignored and the application of the scientific empirical approach to reduce uncertainties is limited in WSNs.

In Chapter 6, Systematic Protocol Evaluation Technique (SPET) is proposed to evaluate and validate the performance of a network protocol. Within the SPET, a Conceptual Statistical Test Framework (CSTF) is constructed using statistical tests to reduce statistical variation errors in the experiments. SPET uses extensive testing in simulation to confirm that the simulation is correctly conducted and the obtained results are a valid representation of the real hardware. The real hardware testing allows us to confirm the trends of simulation and understand the degree of similarity between the two.

Although the results have shown that the reality gap between the hardware and simulation exists, there was a degree of similarity observed in the results. SPET has provided us more confidence in the results observed in a WSN's experiment by incorporating a comprehensive empirical approach with more test cases and statistical techniques to evaluate a WSNs protocol. The results have shown that the MRP can achieve a significantly better performance than AODV and NST.

## 7.2 Revisiting the research questions

The main goal of this thesis is to show the dependability of the WSNs can be achieved through the development and evaluation of a dependable routing protocol. To achieve the goal, we hypothesise in Section 1.2 of Chapter 1 that *a multi-modal approach with an immune-inspired classifier can achieve a better dependability in term of packet delivery rate with a lower energy consumption than a single-mode routing protocol. We also motivated the need to identify the network characteristics in order for an appropriate routing mode to be applied to achieve dependability. It is known that the immune systems exhibit several similar characteristics and properties to WSNs that can the*

*fault tolerance to achieve dependability. Hence, the AIS is applied to our work to identify the failure characteristics and assist the MRP in recovery to improve the dependability.*

*In order to show that the goal has been achieved, we revisit and answer the research questions formulated in Chapter 1.*

- Can we improve the dependability of WSNs by integrating and switching between different routing protocols using a multi-modal approach to function according to its operating environments?
  *To answer the first research question, it is necessary to select appropriate protocols to operate in the multi-modal mode. Chapter 3 presented the research methodology to evaluate the performance of different routing protocols. The routing protocol is chosen as the subject as it plays an important role in achieving dependability and is susceptible to failures triggered by the radio irregularity. To establish the impact of link stability on the routing protocol and identify the routing protocols to operate in multi-modal mode, Chapter 3 investigated the performance of different WSNs protocols and three routing protocols namely AODV, NST, and TinyAODV were identified. Using the three routing protocols identified, the MRP is proposed. The robustness and scalability of MRP are evaluated against single mode routing (AODV and NST) in Chapter 4. The results have shown that the MRP has a higher PDR and lower energy consumption, routing overhead and delay than AODV and NST. The statistical tests have shown that the differences are significant.*

- Research Question 2: Can immune-inspired algorithm be applied to assist the routing protocol to classify the different characteristics of the non-intentional interference in order to improve the response?
  *From many different AIS algorithm proposed in the literature, we have identified in Chapter 5 that the RDA can detect changes in a dynamic environment. Our application of RDA was to determine the characteristics of the interference and classify them according to the strength and duration. The RDA is extended in order to perform classification and interact with the MRP to provide response that can rectify the failures effectively. IDRS is proposed and evaluated in both real hardware and software. The results have shown that the RDA can assist the MRP to provide a good response that has effective increased the packet delivery reliable and reduce communication overhead.*

- Research Question 3: Can the dependability of network protocol be demonstrated by reducing the experimental uncertainty using state of the art statistical techniques? *The statistical techniques are usually used to show the confidence and reduce the error observed in measurement. As the results observed from the WSNs simulation is always different from the hardware, there are two ways*

*where the statistical approaches can be used to demonstrate the dependability of the protocols under evaluation. First, a systematic approach with a statistical approach to the design of the experiments allows us to show the correctness of the experiment and reduce the epistemic and aleatory uncertainties. Second, the application of the statistical tests on the data allows us to measure the significant of the measurements. To address the final research question, the SPET developed on the statistical techniques is proposed and evaluated in Chapter 6. From the presented results, SPET has provided the evidences to show that MRP are significantly better than AODV and NST and the experiment has been performed correctly and not biased toward particular protocol under evaluation.*

*Based on the results and the findings from the works investigated and discussed in this thesis, it can be concluded that the multi-modal recovery approach with the AIS improve the dependability of the WSNs.*

## 7.3 Recommendations for further research

*During the evaluation and analysis of this thesis, several opportunities have emerged for derivative and future research.*

***Implementation in a real operating environment in large scale***

*Although the proposed scheme and strategies have been extensively analysed and evaluated, the number of nodes deployed may not be large enough to represent the real world deployment. An attempt has been made to produce a test environment as close to the real where a small number of the real hardware nodes has been deployed to test the IDRS using interference from a laptop. The scalability of the IDRS is also evaluated on simulation using interference pattern captured on the real hardware motes. Although the results from both hardware and simulation have shown significant improvement in dependability, there is still a need to evaluate the IDRS in a real operating environment where a higher number of nodes can be deployed. However, a lot of time and effort is needed to prepare and select an appropriate testing environment and procure the sensor motes. The experiment may need to be run for a long period of time (months) before any useful and significant results can be observed. As a result, one of the immediate future works is to test the IDRS using a large number of nodes in a building with many other radio emitting devices that can disrupt the communication between nodes.*

***Ability to classify more faults***

*The work in this thesis has been tested on irregular interferences generated by wireless Internet devices. It is known that there are also other radio emitting devices that can interrupt the low power communication of the WSNs node. Hence, it would be interesting to investigate whether the proposed IDRS can still tolerate failures that are generated by a combination of the different radio interference devices where the interference pattern generated can be permanent or transients. In principle, the proposed scheme should still work as the interference generated by IEEE-802.11 devices is more complex and unpredictable. Even if there is an interference pattern that is different from IEEE-802.11, the RDA should be able to learn and detect it. However, it will be interesting to see is there any interference that can fail IDRS or at what point will the RDA fail to distinguish the interference.*

***Optimisation of the algorithm through parameter analysis***

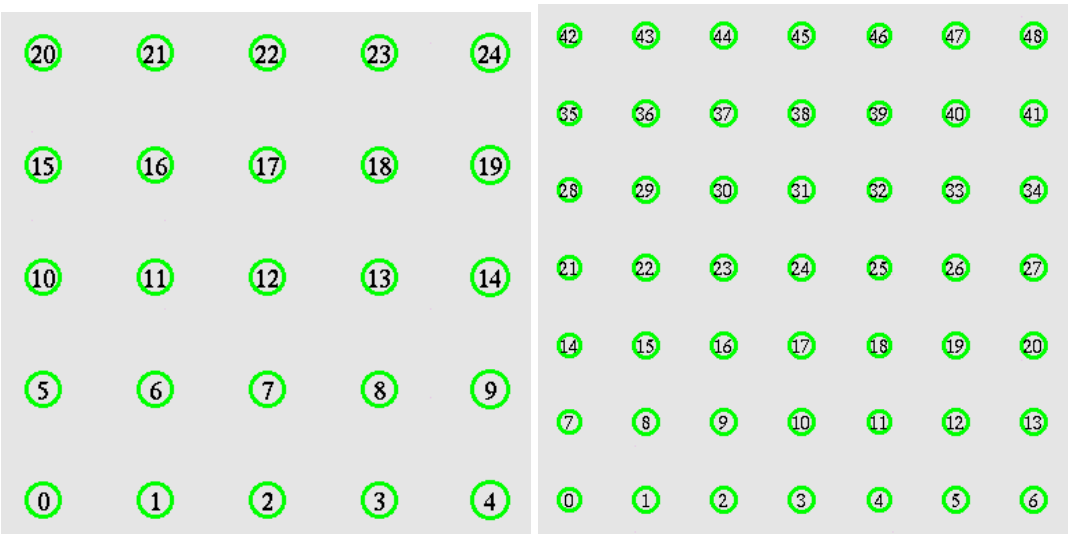
*There are a few parameters in the RDA that can affect the detection rate of the IDRS. Using an iterative approach, a small set of training data is used to determine the parameters that can yield a high true positive detection and recovery rate. Although this approach may not give the best parameter with an optimum result, it is still able to detect and recovery from the interference with a high accuracy. With the availability of the complex search algorithms, it would not be difficult to establish the optimum parameters offline. However, it is more challenging to explore whether it is possible apply reinforcement learning to tune the parameters online in order to achieve optimum solution. With the ability of the MRP to interact and work in parallel with RDA and provide feedback to each other, it will be an interesting research area to investigate whether these parameters tuning can be automated to achieve the optimum performance and adapt to the environment.*

APPENDIX

# A

# Screenshoot for Grid Topology

*The gird topology used to evaluate the routing protocols presented in Chapter 4.*




(a) Network Topology 5 by 5 (b) Network Topology 7 by 7

Figure A.1: Network Topology 5 by 5 (Figure A.1(a)) and 7 by 7 (Figure A.1(b) )

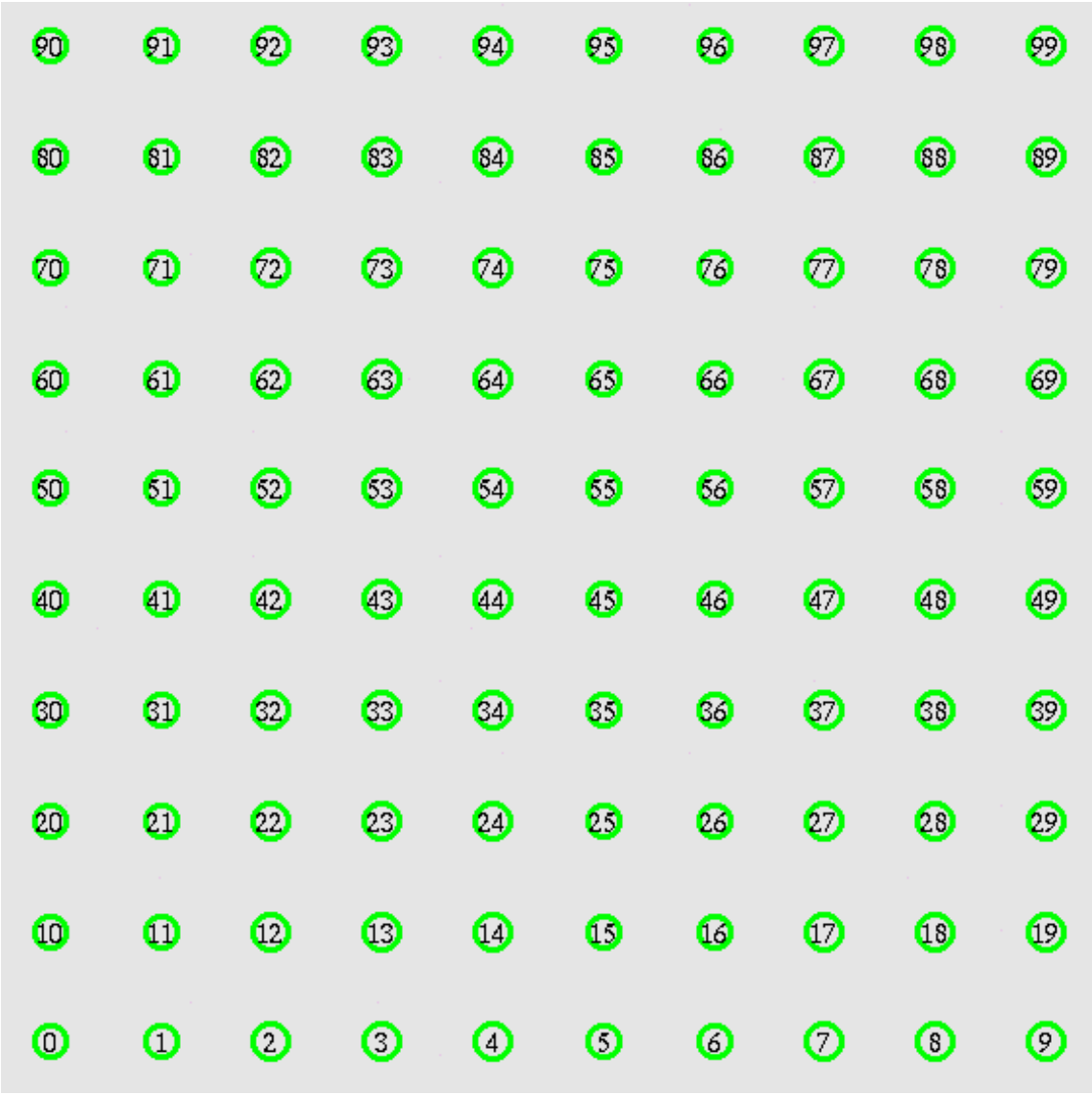



Figure A.2: Network Topology 10 by 10

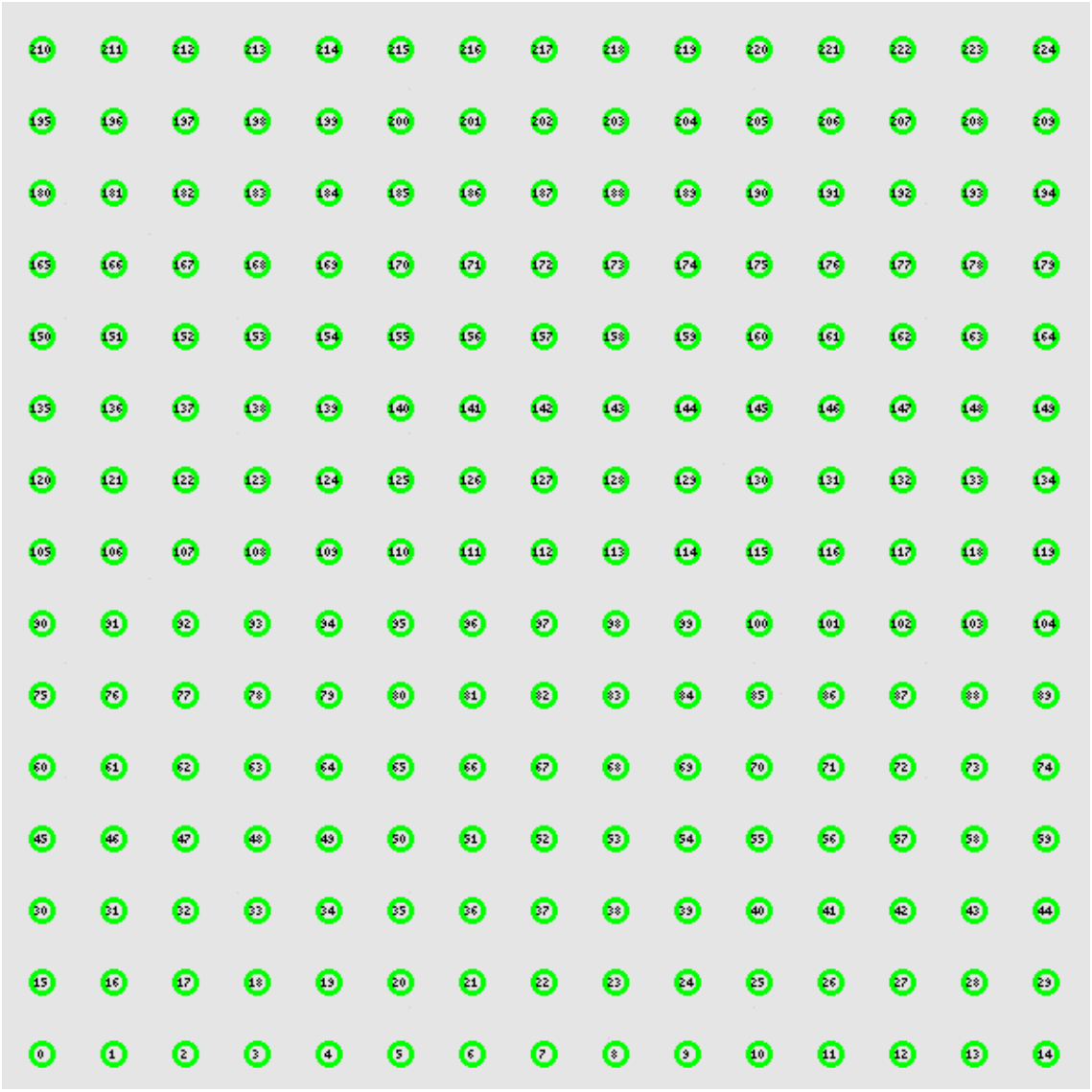



Figure A.3: Network Topology 15 by 15

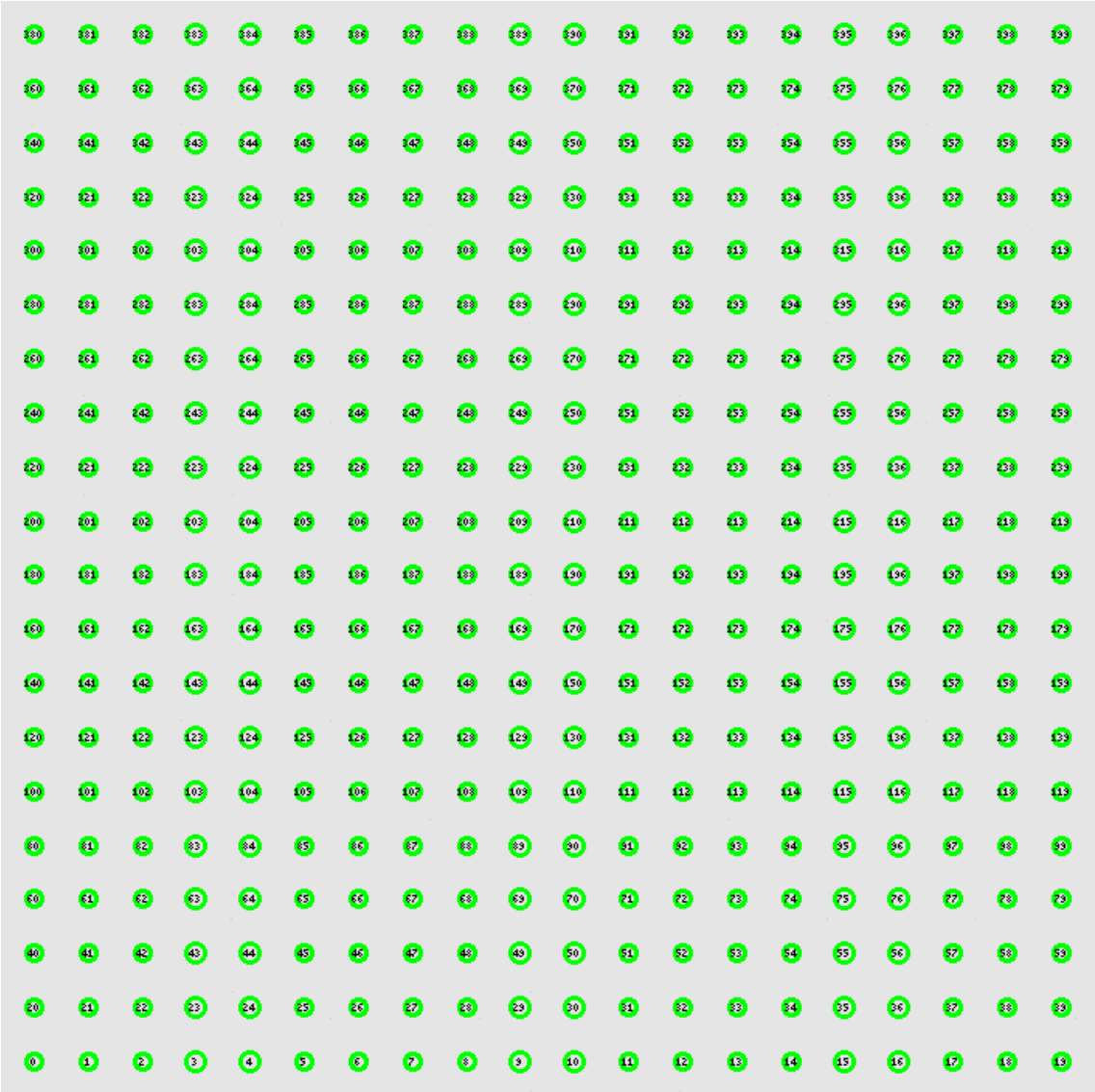


Figure A.4: Network Topology 20 by 20

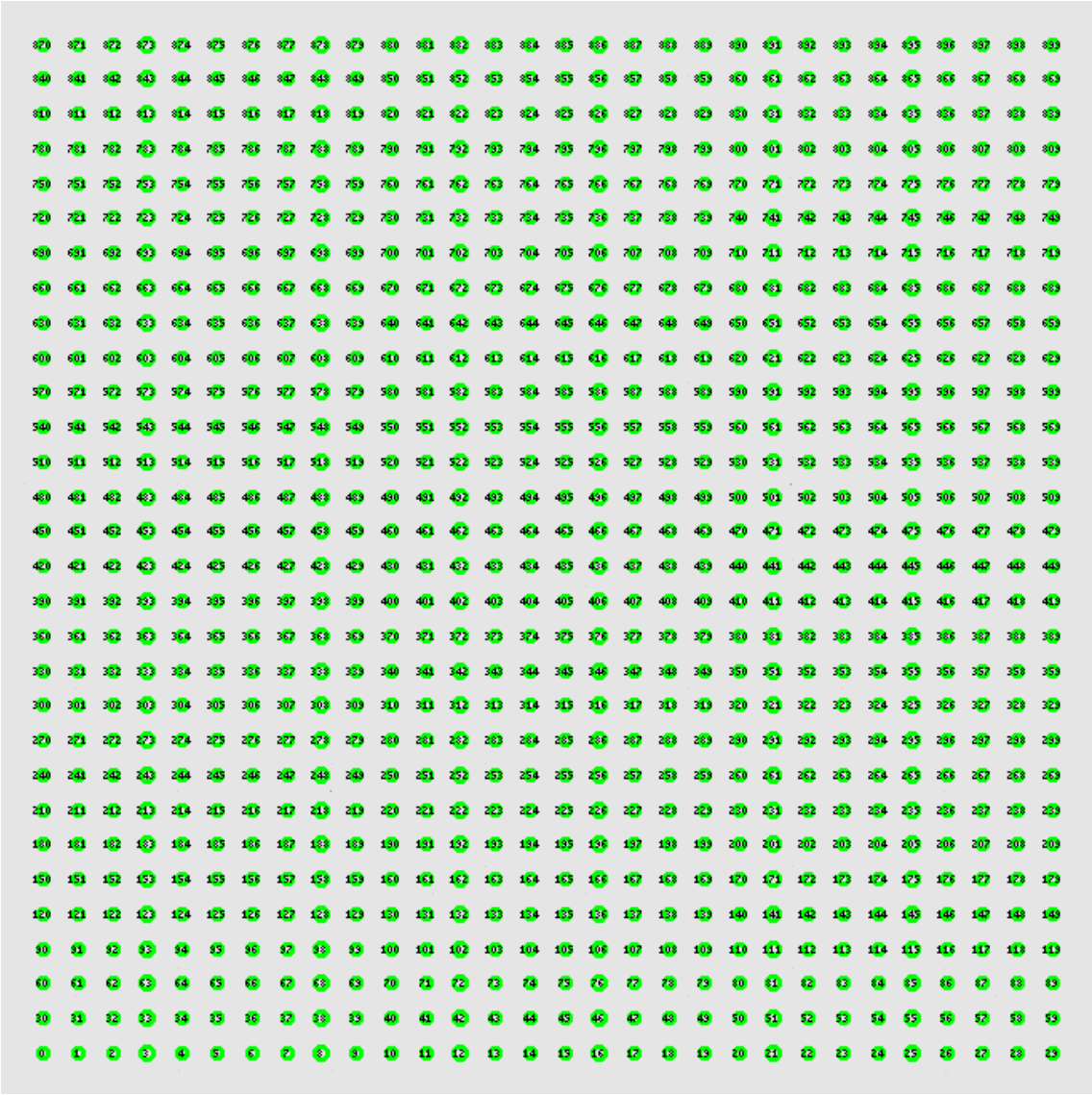


Figure A.5: Network Topology 30 by 30

# List of Abbreviations

*4C - Correctness, Consistency, Completeness and Cost*
*ACK - Acknowledgement*
*ADS - Anomaly Detection Systems*
*ANN - Artificial Neural Networks*
*AIS - Artificial Immune Systems*
*AODV - Adhoc On-demand Distance Vector*
*AOMDV - Adhoc On-demand Multi-path Distance Vector*
*APC - Antigen Presenting Cell*
*ANN - Artificial Neural Networks*
*B-MAC - Berkeley MAC*
*BER - Bit Error Rate*
*CBR - Constant Bit Rate*
*CCA - Clear Channel Assessment*
*CH - Cluster Head*
*CIS - Cognitive Immune Systems*
*CS - Carrier Sensing*
*CSMA - Carrier Sense Multiple Access*
*CSMA/CA - Carrier Sense Multiple Access with Collision Avoidance*
*CSTF - Conceptual Statistical Test Framework*
*CTS - Clear To Send*
*DC - Dendritic Cell*
*DCA - Dendritic Cell Algorithm*
*DSR - Dynamic Source Routing*
*ED - Energy Detection*
*ESRT - Event-to-Sink Reliable Transport*
*FTP - File Transfer Protocol*
*GD - Global Discovery*
*HTTP - HyperText Transfer Protocol*
*IDRS - Immune-inspired Detection and Recovery Systems*
*IEEE - Institute of Electrical and Electronics Engineers*

*KNN - K-Nearest Neighbour*
*LD - Local Discovery*
*LLN - Link Layer Notification*
*LOS - Line of Sight*
*LPL - Low Power Listening*
*LQI - Link Quality Indicator*
*MAC - Medium Access Control*
*MASS - Mobile Adhoc and Sensor Systems*
*MHC - Major Histocompatibility Complex*
*MRP - Multi-modal Routing Protocol*
*MTTF - Mean Time To Failure*
*MTTR - Mean Time To Repair*
*N-ACK - Negative Acknowledgement*
*NS - Network Simulator*
*NSA - Negative Selection Algorithm*
*OS - Operating Systems*
*PAMP - Pathogen associated molecular pattern*
*PDR - Packet Deliver Ratio*
*PER - Packet Error Rate*
*PHY - Physical Layer*
*PRR - Packet Reception Ratio*
*RAM - Random Access Memory*
*RD - Route Discovery*
*RDA - Receptor Density Algorithm*
*RERR - Route Error*
*RF - Radio Frequency*
*RFI - Radio Frequency Interference*
*RREQ - Route Request*
*RREP - Route Response*
*RSSI - Received Signal Strength Indicator*
*RTS - Request To Send*
*SCS - Safety Critical Systems*
*SPET - Systematic Protocol Evaluation Technique*
*SVM - Support Vector Machines*
*TCR - T-cell Receptor*
*TDMA - Time Domain Multiple Access*
*WLAN - Wireless Local Area Network*
*WPAN - Wireless Personal Area Network*
*WSN - Wireless Sensor Networks*

# Glossary

*This glossary provides definitions related to the WSNs and biological terminology. These definitions are taken from the following references: Goldsby et al. (2003) and De Castro and Timmis (2002).*

***Adaptive immune system/Adaptive immunity**: A system is composed of highly specialized, systemic cells and processes that eliminate or prevent pathogenic attack.*

***Affinity**: The strength of binding interaction between antigen and antibody molecules.*

***Antibody**: Protein molecule produced and secreted by B-Cell in response to antigen.*

***Antigen**: Substance that can induce an immune response when introduced into the body.*

***Antigen-presenting cell**: A cell that can "present" antigen in a form that T cells can recognize it.*

***Autoimmune response**: A response that occurs when the body tissues are attacked by its own immune system.*

***B-cells**: A type of lymphocyte (white blood cell) that can mature into a plasma cells and memory B cells.*

***Bone marrow**: The soft blood-forming tissue that fills the cavities of bones and contains fat and immature and mature blood cells, including white blood cells, red blood cells, and platelets.*

***Co-stimulation**: An event in the immune system involving the delivery of a second signal by an antigen-presenting cell.*

***Cytokine**: A small protein released by cells that has a specific effect on the interactions between cells, on communications between cells or on the behavior of cells.*

***Danger signal**: Signals produced as a result of premature or unplanned cell death.*

***Dendritic cell**: A special type of cell that is a key regulator of the immune system, acting as a professional antigen-presenting cell (APC) capable of activating nave T cells and stimulating the growth and differentiation of B cells.*

***Differentiation (in cells)**: A process where a less specialized (premature) cell becomes a more specialized cell type.*

***ECG**: ElectroCardioGram (EKG) is a noninvasive test that is used to reflect underlying heart conditions by measuring the electrical activity of the heart.*
***Effector**: An cell capable of responding to a stimulus.*
***Effector function**: A function that transmits an impulse to immune system.*
***EMG**: ElectroMyoGram (EMG) is a test that is used to record and detect abnormal electrical activity of muscle occur in many diseases and conditions.*
***Enzymes**: are molecules that can cause and accelerate chemical reactions.*

***Homeostasis**: A property of cells, tissues, and organisms that allows the maintenance and regulation of the stability and constancy needed to function properly.*

***Innate immune system/Innate immunity**: The innate immune system comprises the cells and mechanisms that defend the host from infection by other organisms, in a non-specific manner.*

***Lymph nodes**: Small rounded or bean-shaped masses of lymphatic tissue surrounded by a capsule of connective tissue, that are located in many places. in the lymphatic system throughout the body. Lymph nodes filter the lymphatic fluid and store special cells that can remove bacteria and proteins that are traveling through the body in the lymph fluid.*
***Lymphocytes**: A small white blood cell that plays a large role in defending the body against disease and are responsible for immune responses.*

***Macrophage**: A type of white blood that ingests (takes in) foreign material.*
***Major histocompatibility complex**: A cluster of genes encoding polymorphic cell-surface molecules concerned with antigen production and critical to the success of transplantation.*
***Memory cells**: Mature B cells that have an affinity for a particular antigen, where a second exposure with that antigen leads to an enhanced and faster response.*
***Micro-organism**: A very tiny living object.*

***Pathogen**: A microorganism that can cause illness such as bacteria, viruses, and fungi.*
***Pathogen associated molecular pattern**: Pathogen associated molecular pattern (PAMP) are proteins expressed exclusively by pathogen, which can be detected by DCs and result in immune activation.*
***Plasma cells**: Mature B Cells that produce antibodies (proteins) necessary to fight off infections.*
***Proteins**: Organic compounds made up of amoni acid found in animal cells.*

***Receptor**: A molecule on the surface of a cell that selectively receives and binds a specific substance.*

***Somatic mutation**: A process that occurs during clonal expansion that permits refinement of the antibody specificity with relation to the selective antigen.*

***Stem cells**: Primitive, "unspecialized" cells that are able to divide and become specialized cells of the immune systems.*

***T-cells**: A type of lymphocytes, that attack body cells when they have been taken over by viruses or have become cancerous.*
***T-helper cell**: A type of T cell that provides help to other cells in the immune response by recognizing foreign antigens and secreting substances called cytokines that activate T and B cells.*
***Thymus**: A lymphoid organ situated in the center of the upper chest just behind the sternum (breastbone) where lymphocytes mature, multiply, and become T cells.*
***White blood cells**: Cells in the immune system that help the body fight infection.*

# Bibliography


*3GPP2, S. (2009). CDMA2000 Evaluation Methodology, Revision A. [Online; accessed 1-February-2013].*

*Abolhasan, M., Wysocki, T., and Dutkiewicz, E. (2004). A review of routing protocols for mobile ad hoc networks.* Ad hoc networks, *2(1):1–22.*

*Aickelin, U., P., B., Cayzer, S., Kim, J., and Mcleodm, J. (2003). Danger Theory: The link between AIS and IDS? In* Proceedings of the 2nd International Conference on Artificial Immune Systems, *pages 147–155.*

*Akkaya, K. and Younis, M. (2005). A survey on routing protocols for wireless sensor networks.* Ad Hoc Networks, *3(3):325–349.*

*Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: a survey.* Computer networks, *38(4):393–422.*

*Akyildiz, I. and Vuran, M. C. (2010).* Wireless Sensor Networks. *John Wiley & Sons, Inc.*

*Al-Karaki, J. N. and Kamal, A. E. (2004). Routing techniques in wireless sensor networks: A survey.* Transactions on Wireless Communications, *11(6):6 – 28.*

*Alshanyour, A. and Baroudi, U. (2008). Bypass AODV: improving performance of ad hoc on-demand distance vector (AODV) routing protocol in wireless ad hoc networks. In* Proceedings of the 1st international conference on Ambient media and systems, *page 17. ICST.*

*Andel, T. and Yasinsac, A. (2006). On the credibility of MANET simulations.* Computer, *39(7):48–54.*

*Anthony, D., Bennett, W. P., Vuran, M. C., Dwyer, M. B., Elbaum, S., Lacy, A., Engels, M., and Wehtje, W. (2012). Sensing through the continent: towards monitoring migratory birds using cellular sensor networks. In* Proceedings of the 11th international conference on Information Processing in Sensor Networks, *pages 329–340. ACM.*

*Arvind, D. and Wong, K. (2004). Speckled computing: Disruptive technology for networked information appliances. In* Proceeding of the International Symposium on Consumer Electronics, *pages 219–223. IEEE.*

*Atakli, I. M., Hu, H. B., Chen, Y., Ku, W. S., and Su, Z. (2008). Malicious node detection in wireless sensor networks using weighted trust evaluation. In* Proceedings of Spring simulation multiconference. *Society for Computer Simulation International.*

*Avizienis, A., Laprie, J., and Randell, B. (2004a). Dependability and its threats: A taxonomy. In Jacquart, R., editor,* Building the Information Society, *volume 156 of* IFIP International Federation for Information Processing, *pages 91–120. Springer US.*

*Avizienis, A., Laprie, J., Randell, B., and Landwehr, A. (2004b). Basic concepts and taxonomy of dependable and secure computing.* Transaction on Dependable Secure Computing, *0(1):11–33.*

*Avizienis, A. and Laprie, J. C. (1986). Dependable computing: From concepts to design diversity.* Proceedings of the IEEE, *74(5):629 – 638.*

*Balakrishnan, H., Padmanabhan, V., Seshan, S., and Katz, R. (1997). A comparison of mechanisms for improving tcp performance over wireless links.* Transactions on Networking, *5(6):756–769.*

*Beckwith, R., Teibel, D., and Bowen, P. (2004). Unwired wine: Sensor networks in vineyards. In* Proceedings of IEEE Sensors, *pages 561–564. IEEE.*

*Bergamini, L., Crociani, C., Vitaletti, A., and Nati, M. (2010). Validation of WSN simulators through a comparison with a real testbed. In* Proceedings of the 7th workshop on Performance evaluation of wireless ad hoc, sensor, and ubiquitous networks, *pages 103–104. ACM.*

*Boers, N. M., Nikolaidis, I., and Gburzynski, P. (2010). Patterns in the RSSI traces from an indoor urban environment. In* Proceedings of International Workshop on Computer Aided Modeling, Analysis and Design of Communication Links and Networks, *pages 61–65. IEEE.*

*Bondavalli, A., Chiaradonna, S., Di Giandomenico, F., and Grandoni, F. (2000). Threshold-based mechanisms to discriminate transient from intermittent faults.* Transactions on Computers, *49(3):230–245.*

*Botta, A., Dainotti, A., and Pescapé, A. (2010). Do you trust your software-based traffic generator?* Communications Magazine, *48(9):158–165.*

*Boughanmi, N. and Song, Y. (2008). A new routing metric for satisfying both energy and delay constraints in wireless sensor networks.* Journal of Signal Processing Systems, *51(2):137–143.*

*Brownfield, M., Fayez, A., Nelson, T., and Davis, N. (2006). Cross-layer wireless sensor network radio power management. In* Wireless Communications and Networking Conference, *volume 2, pages 1160–1165. IEEE.*

*Burgess, M. (1998). Computer immunology. In* Proceedings of the 12th USENIX conference on System administration, *pages 283–298.*

*Burnet, F. (1959).* The clonal selection theory of acquired immunity. *Cambridge University Press.*

*Candea, G., Cutler, J., and Fox, A. (2004). Improving availability with recursive microreboots: a soft-state system case study.* Performance Evaluation, *56(1):213–248.*

*Carbajo, R., Huggard, M., and McGoldrick, C. (2008). An end-to-end routing protocol for peer-to-peer communication in wireless sensor networks. In* Proceedings of the 6th workshop on Middleware for network eccentric and mobile applications, *pages 5–9. ACM.*

*Chakeres, I. and Belding-Royer, E. (2005). Aodv implementation design and performance evaluation.* International Journal of Wireless and Mobile Computing, *2(3):42.*

*Chandola, V., Banerjee, A., and Kumar, V. (2009). Anomaly detection: A survey.* ACM Computing Survey, *41(3):1–58.*

*Chatzigiannakis, V. and Papavassiliou, S. (2007). Diagnosing anomalies and identifying faulty nodes in sensor networks.* Sensors Journal, *7(5):637–645.*

*Chen, B., Muniswamy-Reddy, K., and Welsh, M. (2006). Ad-hoc multicast routing on resource-limited sensor nodes. In* Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality, *pages 87–94. ACM.*

*Chipara, O., Lu, C., Bailey, T., and Roman, G. (2010). Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In* Proceedings of the 8th Conference on Embedded Networked Sensor Systems, *pages 155–168. ACM.*

*Chong, C. and Kumar, S. (2003). Sensor networks: evolution, opportunities, and challenges.* Proceedings of the IEEE, *91(8):1247–1256.*

*Cohen, I. R. (2004).* Tending Adam's Garden : Evolving the Cognitive Immune Self. *Academic Press.*

*Conover, W. (1999). Practical nonparametric statistics.*

*Cui, S., Madan, R., Goldsmith, A., and Lall, S. (2005). Joint routing, MAC, and link layer optimization in sensor networks with energy constraints. In* Proceedings of the International Conference on Communications, *volume 2, pages 725–729. IEEE.*

*da Silva, A. P. R., Martins, M. H. T., Rocha, B. P. S., Loureiro, A. A. F., Ruiz, L. B., and Wong, H. C. (2005). Decentralized intrusion detection in wireless sensor networks. In* Proceedings of the 1st international workshop on Quality of service & security in wireless and mobile networks, *pages 16–23. ACM.*

*Davoudani, D., Hart, E., and Paechter, B. (2007). An immune-inspired approach to speckled computing. In* Artificial Immune Systems, *pages 288–299. Springer.*

*De Castro, L. and Timmis, J. (2002).* Artificial Immune Systems: A New Computational Intelligence Approach*. Springer Verlag.*

*de Castro, L. and von Zuben, F. (2000). An evolutionary immune network for data clustering. In* Proceedings of the 6th Brazilian Symposium on Neural Networks, *page 84.*

*de Castro, L. N. and Von Zuben, F. J. (2002). Learning and optimization using the clonal selection principle.* Transactions on Evolutionary Computation, *6(3):239–251.*

*Demirkol, I., Alagoz, F., Deliç, H., and Ersoy, C. (2006). Wireless sensor networks for intrusion detection: packet traffic modeling.* Communications Letters, *10(1):22–24.*

*D'haeseleer, P. (1996). An immunological approach to change detection: Theoretical results. In* Proceedings of the 9th Computer Security Foundations, *pages 18–27. IEEE.*

*Di Martino, C., Cinque, M., and Cotroneo, D. (2012). Automated generation of performance and dependability models for the assessment of wireless sensor networks.* Transaction on Computer, *61(6):870–884.*

*Drozda, M., Bate, I., and Timmis, J. (2011). Bio-inspired error detection for complex systems. In* Proceedings of the 17th Pacific Rim International Symposium on Dependable Computing, *pages 154–163. IEEE.*

*Drozda, M., Schaust, S., and Szczerbicka, H. (2007). AIS for misbehavior detection in wireless sensor networks: Performance and design principles. In* Proceedings of the Congress on Evolutionary Computation, *pages 3719 –3726. IEEE.*

*Dunia, R., Qin, S., Edgar, T., and McAvoy, T. (1996). Identification of faulty sensors using principal component analysis.* AIChE Journal, *42(10):2797–2812.*

*Dwivedi, A. K. and Vyas, O. P. (2011). An exploratory study of experimental tools for wireless sensor networks.* Wireless Sensor Network, *3(7):215–240.*

*Elberfeld, M. and Textor, J. (2011). Negative selection algorithms on strings with efficient training and linear-time classification.* Theoretical Computer Science, *412(6):534–542.*

*Elnahrawy, E. and Nath, B. (2003). Cleaning and querying noisy sensors. In* Proceedings of the 2nd international conference on Wireless sensor networks and applications, *pages 78–87. ACM.*

*Fang, L. and Lin, L. (2005). Unsupervised anomaly detection based on an evolutionary artificial immune network. In* Proceedings of Applications of Evolutionary Computing, *volume 3449, pages 166–174.*

*Figueredo, C., Santos, A., Loureiro, A., and Nogueira, J. (2005). Policy-based adaptive routing in autonomous wsns.* Ambient Networks, *775:206–219.*

*Fonseca, R., Gnawali, O., Jamieson, K., Kim, S., Levis, P., and Woo, A. (2006). The collection tree protocol (CTP). http://www.tinyos.net/tinyos-2.x/doc/html/tep123.html. [Online; accessed 1-February-2013].*

*Forrest, S., Perelson, A., Allen, L., and Cherukuri, R. (1994). Self-nonself discrimination in a computer. In* Proceedings of the Symposium on Research in Security and Privacy, *pages 202–212. IEEE.*

*Gama, O., Carvalho, P., and Mendes, P. (2011). A model to improve the accuracy of wsn simulations. In* Wired/Wireless Internet Communications, *pages 128–139. Springer.*

*Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., and Levis, P. (2009). Collection tree protocol. In* Proceedings of the 7th Conference on Embedded Networked Sensor Systems, *pages 1–14. ACM.*

*Goldsby, R., Kindt, T., and Osborne, B. (2003).* Kuby's Immunology. *W. H. Freeman.*

*Gomez, C., Salvatella, P., Alonso, O., and Paradells, J. (2006). Adapting AODV for IEEE 802.15.4 mesh sensor networks: Theoretical discussion and performance evaluation in a real environment. In* Proceedings of the International Symposium on on World of Wireless, Mobile and Multimedia Networks, *pages 159–170. IEEE.*

*Görnitz, N., Kloft, M., Rieck, K., and Brefeld, U. (2013). Toward supervised anomaly detection.* Journal Artificial Intelligence Research, *46:235–262.*

*Gravetter, F. (2012).* Essentials of statistics for the behavioral sciences. *Cengage Learning.*

*Greensmith, J., Aickelin, U., and Cayzer, S. (2005). Introducing dendritic cells as a novel immune-inspired algorithm for anomaly detection. In* Proceedings of the 4th International Conference on Artificial Immune Systems, *pages 153–167.*

*Grossman, Z. and Paul, W. (2001). Autoreactivity, dynamic tuning and selectivity.* Current opinion in immunology, *13(6):687–698.*

*Gungor, V. and Hancke, G. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches.* Transactions on Industrial Electronics, *56(10):4258–4265.*

*Gupta, S., Zheng, R., and Cheng, A. (2007). ANDES: An anomaly detection system for wireless sensor networks. In* Proceeding of International Conference on Mobile Adhoc and Sensor Systems*, pages 1–9. IEEE.*

*Gutierrez, J., Naeve, M., Callaway, E., Bourgeois, M., Mitter, V., and Heile, B. (2001). IEEE 802.15.4: a developing standard for low-power low-cost wireless personal area networks.* Network*, 15(5):12–19.*

*Haas, Z. and Pearlman, M. (2001). The performance of query control schemes for the Zone Routing Protocol (ZRP).* Transactions on Networking*, 9(4):427–438.*

*Hai, T., Huh, E., and Jo, M. (2010). A lightweight intrusion detection framework for wireless sensor networks.* Wireless Communications and Mobile Computing*, 10(4):559–572.*

*Hai, T., Khan, F., and Huh, E. (2007). Hybrid intrusion detection system for wireless sensor networks.* Computational Science and Its Applications*, 0(0):383–396.*

*Hart, E. and Timmis, J. (2008). Application areas of AIS: The past, the present and the future.* Applied Software Computing*, 8(1):191–201.*

*Hatler, M. (2012). Industrial wireless sensor networks: Trends and developments. [Online; accessed 1-February-2013].*

*Hawrylak, P. J., Cain, J. T., and Mickle, M. H. (2009). Analysis methods for sensor networks. In Misra, S. C., Woungang, I., and Misra, S., editors,* Guide to Wireless Sensor Networks*, Computer Communications and Networks, pages 635–658. Springer London.*

*Heinzelman, W. R., Chandrakasan, A., and Balakrishnan, H. (2000). Energy-efficient communication protocol for wireless microsensor networks. In* Proceedings of the 33rd Hawaii International Conference on System Sciences*, volume 2, pages 3005–3014.*

*Helton, J. (1997). Uncertainty and sensitivity analysis in the presence of stochastic and subjective uncertainty.* Journal of Statistical Computation and Simulation*, 57(1-4):3–76.*

*Helton, J. (2008). Uncertainty and sensitivity analysis for models of complex systems. In Graziani, F., editor,* Computational Methods in Transport: Verification and Validation*, volume 62 of* Lecture Notes in Computational Science and Engineering*, pages 207–228. Springer Berlin Heidelberg.*

*Hida, Y., Huang, P., and Nishtala, R. (2004). Aggregation query under uncertainty in sensor networks. Technical report, Department of Electrical Engineering and Computer Science, University of California.*

*Hilder, J., Owens, N., Neal, M., Hickey, P., Cairns, S., Kilgour, D., Timmis, J., and Tyrrell, A. (2012). Chemical detection using the receptor density algorithm.* Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 42(6):1730–1741.*

*Hodge, V. and Austin, J. (2004). A survey of outlier detection methodologies.* Artificial Intelligence Review, *22(2):85–126.*

*Hofmeyr, S. and Forrest, S. (2000). Architecture for an artificial immune system.* Evolutionary Computing, *8(4):443–473.*

*Howitt, I. and Gutierrez, J. (2003). IEEE 802.15.4 low rate - wireless personal area network coexistence issues. In* Proceeding of the conference of Wireless Communications and Networking, *volume 3, pages 1481 –1486. IEEE.*

*Hsu, L., King, C., and Banerjee, A. (2007). On broadcasting in wireless sensor networks with irregular and dynamic radio coverage. In* International Conference on Parallel Processing, *pages 55–55. IEEE.*

*Huo, H., Xu, Y., Bilen, C., and Zhang, H. (2009). Coexistence issues of 2.4GHz sensor networks with other RF devices at home. In* Proceeding of the 3rd International Conference on Sensor Technologies and Applications, *pages 200–205. IEEE.*

*IEEE, C. S. (2006a). Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf. [Online; accessed 1-March-2013].*

*IEEE, C. S. (2006b). Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). [Online; accessed 1-April-2013].*

*Ihaka, R. and Gentleman, R. (1996). R: A language for data analysis and graphics.* Journal of computational and graphical statistics, *5(3):299–314.*

*Ingelrest, F., Barrenetxea, G., Schaefer, G., Vetterli, M., Couach, O., and Parlange, M. (2010). Sensorscope: Application-specific sensor network for environmental monitoring.* Transactions on Sensor Networks, *6(2):17.*

*Ioannis, K., Tassos, D., and Felix, C. F. (2007). Towards intrusion detection in wireless sensor networks. In* Proceedings of the 13th European Wireless Conference, *pages 1–10.*

*ISA100, S. C. (2009). ISA100.11a, Wireless Systems for Industrial Automation: Process Control and Related Applications. http://www.isa.org.*

*Ishida, Y. (1997). Active diagnosis by self-organization: An approach by the immune network metaphor. In* Proceedings of the 15th international joint conference on Artifical intelligence, *pages 1084–1089.*

*Iyer, Y., Gandham, S., and Venkatesan, S. (2005). STCP: A generic transport layer protocol for wireless sensor networks. In* Proceedings of 14th International Conference on Computer Communications and Networks, *pages 449 – 454.*

*Janeway Jr, A. (1992). The immune system evolved to discriminate infectious nonself from non-infectious self.* Immunology Today, *13(1):11 – 16.*

*Jerne, N. (1974). Towards a network theory of the immune system.* Annual Immunology Annales d'immunologie, *125C:373–389.*

*Johnson, D. and Maltz, D. (1996). Dynamic source routing in ad hoc wireless networks.* Kluwer International Series in Engineering and Computer Science, *pages 153–179.*

*Kahn, J. M., Katz, R. H., and Pister, K. S. J. (1999). Next century challenges: mobile networking for Smart Dust. In* Proceedings of the 5th annual international conference on Mobile computing and networking, *pages 271–278. ACM.*

*Karlof, C. and Wagner, D. (2003). Secure routing in wireless sensor networks: Attacks and countermeasures. In* Proceedings of the 1st International Workshop on Sensor Network Protocols and Applications., *pages 113 – 127. IEEE.*

*Kessler, G. and Shepard, S. (1997). A Primer On Internet and TCP/IP Tools and Utilities. RFC 2151. [Online; accessed 1-February-2013].*

*Kim, J. and Bentley, P. (2001a). An evaluation of negative selection in an artificial immune system for network intrusion detection. In* Proceedings of the Genetic and Evolutionary Computation Conference, *pages 1330–1337.*

*Kim, J. and Bentley, P. (2001b). Towards an artificial immune system for network intrusion detection: An investigation of clonal selection with a negative selection operator. In* Proceedings of the Congress on Evolutionary Computation, *volume 2, pages 1244 –1252. IEEE.*

*Kim, J. and Bentley, P. (2002). Towards an artificial immune system for network intrusion detection: An investigation of dynamic clonal selection. In* Proceedings of the Congress on Evolutionary Computation, *pages 1015–1020. IEEE.*

*Ko, J. and Terzis, A. (2010). Power control for mobile sensor networks: An experimental approach. In* Proceedings of the 7th Annual Communications Society Conference on Sensor Mesh and Ad Hoc Communications and Networks. *IEEE.*

*Kulakov, A. and Davcev, D. (2005). Tracking of unusual events in wireless sensor networks based on artificial neural-networks algorithms. In* Proceedings of the International Conference on Information Technology: Coding and Computing, *volume 2, pages 534–539. IEEE.*

*Kurkowski, S., Camp, T., and Colagrosso, M. (2005). MANET simulation studies: the incredibles.* SIGMOBILE Mobile Computing and Communications Review, *9(4):50–61.*

*Lamport, L., Shostak, R., and Pease, M. (1982). The Byzantine Generals problem.* Transactions on Programming Languages and Systems, *4(3):382–401.*

*Langendoen, K. (2008). Medium access control in wireless sensor networks.* Medium access control in wireless networks, *2:535–560.*

*Langendoen, K., Baggio, A., and Visser, O. (2006). Murphy loves potatoes: Experiences from a pilot sensor network deployment in precision agriculture. In* Proceedings of the 20th international conference on Parallel and distributed processing, *pages 174–174. IEEE Computer Society.*

*Laprie, J. (1985). On Computer System Dependability: Faults, Errors, and Failures. In* Proceeeding of the 13th Computer Society International Conference, *pages 256–259. IEEE.*

*Lau, H. K., Timmis, J., and Bate, I. (2011). Collective self-detection scheme for adaptive error detection in a foraging swarm of robots. In* the 10th international conference on Artificial immune systems, *pages 254–267.*

*Le Boudec, J. and Sarafijanovic, S. (2004). An artificial immune system approach to misbehavior detection in mobile ad-hoc networks. In* Proceedings of the First International Workshop on Biologically Inspired Approaches to Advanced Information Technology, *pages 96–111.*

*Lédeczi, A., Nádas, A., Völgyesi, P., Balogh, G., Kusy, B., Sallai, J., Dóra, S., Molnár, K., Maróti, M., and Simon, G. (2005). Countersniper system for urban warfare.* Transactions on Sensor Networks, *1(2):153–177.*

*Lee, H., Cerpa, A., and Levis, P. (2007a). Improving wireles simulation through noise modeling. In* Proceedings of the 6th international conference on Information processing in sensor networks, *pages 21–30. ACM.*

*Lee, S., Jeon, T., Hwang, H., and Kim, C. (2007b). Design and implementation of wireless sensor based-monitoring system for smart factory. In* Computational Science and Its Applications, *pages 584–592. Springer.*

*Leentvaar, K. and Flint, J. (1976). The capture effect in FM receivers.* Transactions on Communications, *24(5):531–539.*

*Lin, S., Zhang, J., Zhou, G., Gu, L., Stankovic, J., and He, T. (2006). ATPC: adaptive transmission power control for wireless sensor networks. In* the 4th international conference on Embedded networked sensor systems, *pages 223–236.*

*Lin, S., Zhou, G., Whitehouse, K., Wu, Y., Stankovic, J., and He, T. (2009). Towards stable network performance in wireless sensor networks. In* Proceedings of the 30th Real-Time Systems Symposium, *pages 227–237. IEEE.*

*Liu, H., Li, J., Xie, Z., Lin, S., Whitehouse, K., Stankovic, J. A., and Siu, D. (2010). Automatic and robust breadcrumb system deployment for indoor firefighter applications. In* Proceedings of the 8th international conference on Mobile systems, applications, and services*, pages 21–34. IEEE.*

*Liu, H., Xie, Z., Li, J., Lin, S., Siu, D., Hui, P., Whitehouse, K., and Stankovic, J. (2013). An automatic, robust, and efficient multi-user breadcrumb system for emergency response applications.* Transactions on Mobile Computing*, 99(1):1 – 14.*

*Loo, C., Ng, M., Leckie, C., and Palaniswami, M. (2006). Intrusion detection for routing attacks in sensor networks.* International Journal of Distributed Sensor Networks*, 2(4):313–332.*

*Luo, X., Dong, M., and Huang, Y. (2006). On distributed fault-tolerant detection in wireless sensor networks.* Transactions on Computers*, 55(1):58–70.*

*Mahanti, A., Carlsson, N., Williamson, C., and Arlitt, M. (2010). Ambient interference effects in Wi-Fi networks. In* Networking*, pages 160–173. Springer.*

*Mainwaring, A., Culler, D., Polastre, J., Szewczyk, R., and Anderson, J. (2002). Wireless sensor networks for habitat monitoring. In* Proceedings of the 1st international workshop on Wireless sensor networks and applications*, pages 88–97. ACM.*

*Mal-Sarkar, S., Sikder, I., Yu, C., and Konangi, V. (2009). Uncertainty-aware wireless sensor networks.* International Journal of Mobile Communications*, 7(3):330–345.*

*Malan, D., Fulford-Jones, T., Welsh, M., and Moulton, S. (2004). CodeBlue: An Ad Hoc Sensor Network Infrastructure for Emergency Medical Care. In* Proceedings of the workshop on Applications of Mobile Embedded Systems*, pages 12–14. ACM.*

*Marina, M. K. and Das, S. R. (2002). Ad hoc on-demand multipath distance vector routing.* SIGMOBILE Mobile Computing Communication Review*, 6:92–93.*

*Matzinger, P. (1994). Tolerance, danger, and the extended family.* Annual review of immunology*, 12(1):991–1045.*

*Mosmann, T. and Livingstone, A. (2004). Dendritic cells: the immune information management experts.* Nature Immunology*, 5(1):564–566.*

*Moyne, J. and Tilbury, D. (2007). The emergence of industrial control networks for manufacturing control, diagnostics, and safety data.* Proceedings of the IEEE*, 95(1):29–47.*

*Murphy, K., Travers, P., and Walport, M. (2012).* Janeway's immunobiology*, volume 7. Garland Science New York, NY, USA.*

*Nam, D. H. and Min, H. K. (2007). An energy-efficient clustering using a round-robin method in a wireless sensor network. In* Proceedings of the 5th ACIS International Conference on Software Engineering Research, Management & Applications*, pages 54–60. IEEE.*

*Ngai, E., Liu, J., and Lyu, M. (2006). On the intruder detection for sinkhole attack in wireless sensor networks. In* The Proceeding of the International Conference on Communications, *volume 8, pages 3383 –3389. IEEE.*

*Ni, K., Ramanathan, N., Chehade, M., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., and Srivastava, M. (2009). Sensor network data fault types.* Transaction in Sensor Networks, *5(3).*

*Noda, C., Prabh, S., Alves, M., Boano, C., and Voigt, T. (2011). Quantifying the channel quality for interference-aware wireless sensor networks.* SIGBED Review, *8(4):43–48.*

*Novak, R. W., Griffin, J., Arif, S., and Mufti, A. (2006).* Immunology. *Elsevier Science Limited.*

*NS2, S. (2002). The network simulator ns-2. http://www.isi.edu/nsnam/ns/. [Online; accessed 1-February-2013].*

*Onat, I. and Miri, A. (2005a). An intrusion detection system for wireless sensor networks. In* Proceedings of the international Conference on Wireless And Mobile Computing, Networking And Communications, *volume 3, pages 253–259. IEEE.*

*Onat, I. and Miri, A. (2005b). A real-time node-based traffic anomaly detection algorithm for wireless sensor networks. In* Proceedings of Systems Communications, *pages 422 – 427.*

*Ong, K., Yue, S., and Ling, K. (2010). Implementation of fast fourier transform on body sensor networks. In* International Conference on Body Sensor Networks (BSN), *pages 197–202. IEEE.*

*Owens, N., Greensted, A., Timmis, J., and Tyrell, A. (2009). T cell receptor signalling inspired kernel density estimation and anomaly detection. In* the 8th international conference on Artificial immune systems, *pages 122–135.*

*Owens, N., Greensted, A., Timmis, J., and Tyrrell, A. (2012). The receptor density algorithm.* Theoretical Computer Science.

*Owens, N., Timmis, J., Greensted, A., and Tyrrell, A. (2010). Elucidation of t cell signalling models.* Journal of theoretical biology, *262(3):452–470.*

*Pawlikowski, K., Jeong, H., and Lee, J. (2002). On credibility of simulation studies of telecommunication networks.* Communications Magazine, *40(1):132–139.*

*Perillo, M., Cheng, Z., and Heinzelman, W. (2005). An analysis of strategies for mitigating the sensor network hot spot problem. In* Proceeding on the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, *pages 474–478. IEEE.*

*Perkins, C. and Royer, E. (1999). Ad-hoc on-demand distance vector routing. In* Proceeding of the 2nd Workshop on Mobile Computing Systems and Applications, *pages 90–100. IEEE.*

*Perkins, C., Royer, E., and Das, S. (2003). RFC 3561 Ad hoc On-Demand Distance Vector (AODV) Routing. http://tools.ietf.org/html/rfc3561. [Online; accessed 1-January-2013].*

*Pham, H., Pediaditakis, D., and Boulis, A. (2007). From simulation to real deployments in wsn and back. In* Proceedings of the international Symposium on a World of Wireless, Mobile and Multimedia Networks, *pages 1–6. IEEE.*

*Pirzada, A. and Portmann, M. (2007). High performance aodv routing protocol for hybrid wireless mesh networks. In* Proceedings of the 4th Annual International Conference on Mobile and Ubiquitous Systems: Networking & Services, 2007., *pages 1–5. IEEE.*

*Polastre, J., Szewczyk, R., and Culler, D. (2005). Telos: enabling ultra-low power wireless research. In* Proceeding of the 4th International Symposium on Information Processing in Sensor Networks, *pages 364–369. IEEE.*

*Raghunathan, V., Schurgers, C., Park, S., and Srivastava, M. (2002). Energy-aware wireless microsensor networks.* Signal Processing Magazine, *19(2):40–50.*

*Rajasegarar, S., Leckie, C., and Palaniswami, M. (2006). Distributed anomaly detection in wireless sensor networks. In* Proceedings of the international Conference on Communication, *pages 1–5. IEEE.*

*Rajasegarar, S., Leckie, C., and Palaniswami, M. (2008). Anomaly detection in wireless sensor networks.* Wireless Communications, *15(4):34–40.*

*Rajasegarar, S., Leckie, C., Palaniswami, M., and Bezdek, J. (2007). Quarter sphere based distributed anomaly detection in wireless sensor networks. In* Proceedings of the international Conference on Communications, *pages 3864–3869. IEEE.*

*Rajendran, V., Obraczka, K., Yi, Y., Lee, S., Tang, K., and Gerla, M. (2004). Combining source- and localized recovery to achieve reliable multicast in multi-hop ad hoc networks. In* Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications, *pages 112–124. Springer.*

*Ramanathan, N., Chang, K., Kapur, R., Girod, L., Kohler, E., and Estrin, D. (2005). Sympathy for the sensor network debugger. In* Proceedings of the 3rd international conference on Embedded networked sensor systems, *pages 255–267. ACM.*

*Raymond, D. R. and Midkiff, S. F. (2008). Denial-of-Service in wireless sensor networks: Attacks and defenses.* Transactions on Pervasive Computing, *7(1):74 –81.*

*Read, M., Andrews, P., Timmis, J., and Kumar, V. (2011). Techniques for Grounding Agent-Based Simulations in the Real Domain: a case study in Experimental Autoimmune Encephalomyelitis.* Mathematical and Computer Modelling of Dynamical Systems, *17(4):296–302.*

*Reis, C., Mahajan, R., Rodrig, M., Wetherall, D., and Zahorjan, J. (2006). Measurement-based models of delivery and interference in static wireless networks.* ACM SIGCOMM Computer Communication Review, *36(4):51–62.*

*Roman, R., Zhou, J., and Lopez, J. (2006). Applying intrusion detection systems to wireless sensor networks. In* Proceeding of the 3rd Internation Conference on Consumer Communications and Networking Conference, *volume 1, pages 640 – 644. IEEE.*

*Sankarasubramaniam, Y. and Akan, O.and Akyildiz, I. (2003). ESRT: event-to-sink reliable transport in wireless sensor networks. In* Proceedings of the 4th international symposium on Mobile ad hoc networking & computing, *pages 177–188. ACM.*

*Schaust, S. and Szczerbicka, H. (2011). Applying antigen-receptor degeneracy behavior for misbehavior response selection in wireless sensor networks. In* Proceedings of the 10th international conference on Artificial immune systems, *pages 212–225. Springer Berlin Heidelberg.*

*Schenato, L., Sinopoli, B., Franceschetti, M., Poolla, K., and Sastry, S. (2007). Foundations of control and estimation over lossy networks.* Proceedings of the IEEE, *95(1):163–187.*

*Schoch, E., Feiri, M., Kargl, F., and Weber, M. (2008). Simulation of ad hoc networks: ns-2 compared to jist/swans. In* Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems & workshops, *page 36.*

*Shakshuki, E., Malik, H., and Sheltami, T. (2009). Lessons Learned: Simulation Vs WSN Deployment. In* International Conference on Advanced Information Networking and Applications, *pages 580–587. IEEE.*

*Shapiro, S. and Wilk, M. (1965). An analysis of variance test for normality (complete samples).* Biometrika, *52(3/4):591–611.*

*Siewiorek, D. (1990). Faults and their manifestation. In* Fault-tolerant distributed computing, *pages 244–261. Springer.*

*Sohraby, K., Minoli, D., and Znati, T. (2007).* Wireless Sensor Networks: Technology, Protocols, and Applications*. Wiley-Interscience.*

*Song, J., Han, S., Mok, K., Chen, D., Lucas, M., and Nixon, M. (2008). WirelessHART: Applying wireless technology in real-time industrial process control. In* Real-Time and Embedded Technology and Applications Symposium, *pages 377–386. IEEE.*

*Srinivasan, K., Dutta, P., Tavakoli, A., and Levis, P. (2010). An empirical study of low-power wireless.* Transactions on Sensor Networks, *6(2):1–49.*

*Staddon, J., Balfanz, D., and Durfee, G. (2002). Efficient tracing of failed nodes in sensor networks. In* Proceedings of the 1st international workshop on Wireless sensor networks and applications, *pages 122–130. ACM.*

*Stetsko, A., Stehlík, M., and Matyas, V. (2011). Calibrating and comparing simulators for wireless sensor networks. In* Proceedings of the 8th International Conference on Mobile Ad-hoc and Sensor Systems, *pages 733–738. IEEE.*

*Stibor, T., Timmis, J., and Eckert, C. (2006). On the use of hyperspheres in artificial immune systems as antibody recognition regions. In* Proceedings of 5th International Conference on Artificial Immune Systems, *pages 215–228.*

*Suriyachai, P., Roedig, U., and Scott, A. (2012). A survey of MAC protocols for mission-critical applications in wireless sensor networks.* Communications Surveys & Tutorials, *14(2):240–264.*

*Szewczyk, R., Mainwaring, A., Polastre, J., Anderson, J., and Culler, D. (2004). An analysis of a large scale habitat monitoring application. In* Proceedings of the 2nd international conference on Embedded networked sensor systems, *pages 214–226. ACM.*

*Tanenbaum, A. and Van Steen, M. (2002).* Distributed systems, *volume 2. Prentice Hall.*

*Tate, J. and Bate, I. (2010). Sensornet protocol tuning using principled engineering methods.* The Computer Journal, *53:991–1019.*

*Thelen, J., Goense, D., and Langendoen, K. (2005). Radio Wave Propagation in Potato Fields. In* Proceedings of the First Workshop on Wireless Network Measurements, *volume 2, pages 331–338.*

*Togami, T., Yamamoto, K., Ito, R., Hashimoto, A., and Kameoka, T. (2012). A wireless sensor network for precise soil water management in an orchard. In* International Conference of Argriculture Engineering.

*Tolle, G., Polastre, J., Szewczyk, R., Culler, D., Turner, N., Tu, K., Burgess, S., Dawson, T., Buonadonna, P., Gay, D., and Hong, W. (2005). A macroscope in the redwoods. In* Proceedings of the 3rd international conference on Embedded networked sensor systems, *pages 51–63. ACM.*

*Trinidad, M. and Valle, M. (2009). Reliable event detectors for constrained resources wireless sensor node hardware.* EURASIP Journal on Embedded Systems, *2009:7.*

*Tseng, Y., Ni, S., Chen, Y., and Sheu, J. (2002). The broadcast storm problem in a mobile ad hoc network.* Wireless networks, *8(2/3):153–167.*

*van Hoesel, L. and Havinga, P. (2004). A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing Preamble Transmissions and Transceiver State Switches. In* Proceeding of the 1st International Conference on Networked Sensing Systems*. Society of Instrument and Control Engineers.*

*Vargha, A. and Delaney, H. (2000). A critique and improvement of the CL common language effect size statistics of McGraw and Wong.* Journal of Educational and Behavioral Statistics*, 25(2):101–132.*

*Wagner, R. (2010). Standards-based wireless sensor networking protocols for spaceflight applications. In* Proceedings of the Aerospace Conference*, pages 1–7. IEEE.*

*Wallenta, C., Kim, J., Bentley, P., and Hailes, S. (2010). Detecting interest cache poisoning in sensor networks using an artificial immune algorithm.* Applied Intelligence*, 32(1):1–26.*

*Wan, C., Campbell, A., and Krishnamurthy, L. (2005). Pump-slowly, fetch-quickly (PSFQ): a reliable transport protocol for sensor networks.* IEEE Journal on Selected Areas in Communications*, 23(4):862–872.*

*Wan, C., Eisenman, S., and Campbell, A. (2003). CODA: congestion detection and avoidance in sensor networks. In* Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 266–279. ACM.*

*Wang, P. and Akyildiz, I. (2011). Spatial correlation and mobility-aware traffic modeling for wireless sensor networks.* Transactions on Networking*, 19(6):1860–1873.*

*Wang, Q. and Zhang, T. (2007). Detecting anomaly node behavior in wireless sensor networks. In* Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, volume 1, pages 451–456. IEEE.*

*Wang, S. and Kuo, S. (2003). Communication strategies for heartbeat-style failure detectors in wireless ad hoc networks. In* Proceedings of the International Conference on Dependable Systems and Networks.*, pages 361–370. IEEE.*

*Wang, Y., Attebury, G., and Ramamurthy, B. (2006). A survey of security issues in wireless sensor networks.* IEEE Communications Surveys Tutorials*, 8(2):2 –23.*

*Werner-Allen, G., Lorincz, K., Welsh, M., Marcillo, O., Johnson, J., Ruiz, M., and Lees, J. (2006). Deploying a wireless sensor network on an active volcano.* IEEE Internet Computing*, 10(2):18–25.*

*Whitehouse, K., Woo, A., Jiang, F., Polastre, J., and Culler, D. (2005). Exploiting the capture effect for collision detection and recovery. In* Proceedings of the 2nd workshop on Embedded Networked Sensors*, pages 45–52. IEEE.*

*Wilcoxon, F. (1945). Individual comparisons by ranking methods.* Biometrics bulletin, *1(6):80–83.*

*Woo, A. and Culler, D. (2001). A transmission control scheme for media access in sensor networks. In* Proceedings of the 7th annual international conference on Mobile computing and networking, *pages 221–235.*

*Wu, Y., Kapitanova, K., Li, J., Stankovic, J. A., Son, S. H., and Whitehouse, K. (2010a). Run time assurance of application-level requirements in wireless sensor networks. In* Proceedings of the 9th International Conference on Information Processing in Sensor Networks, *pages 197–208. ACM.*

*Wu, Y., Zhou, G., and Stankovic, J. (2010b). ACR: active collision recovery in dense wireless sensor networks. In* Proceedings of the 29th conference on Information communications, *pages 911–919. IEEE.*

*Xiao, H., Li, T., Ogai, H., Zou, X., Otawa, T., Umeda, S., and Tsuji, T. (2010). The health monitoring system based on distributed data aggregation for WSN used in bridge diagnosis. In* Proceedings of SICE Annual Conference, *pages 2134–2138. IEEE.*

*Xie, M., Han, S., Tian, B., and Parvin, S. (2011). Anomaly detection in wireless sensor networks: A survey.* Journal of Network and Computer Applications, *34(4):1302–1325.*

*Yazir, Y., Farahbod, R., Guitouni, A., Ganti, S., and Coady, Y. (2010). Adaptive routing in mobile ad hoc networks based on decision aid approach. In* Proceedings of the international Workshop on Mobility Management and Wireless Access, *pages 1–10. IEEE.*

*Ye, Y., Heidemann, J., and Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In* Proceedings of 21st Annual Joint Conference of the Computer and Communications Societies, *volume 3, pages 1567–1576. IEEE.*

*You, Z., Zhao, X., Wan, H., Hung, W., Wang, Y., and Gu, M. (2011). A novel fault diagnosis mechanism for wireless sensor networks.* Mathematical and Computer Modelling, *54(1):330–343.*

*Youn, J., Lee, J., Sung, D., and Kang, C. (2006). Quick local repair scheme using adaptive promiscuous mode in mobile ad hoc networks.* Journal of Networks, *1(1):1–11.*

*Younis, M., Youssef, M., and Arisha, K. (2002). Energy-aware routing in cluster-based sensor networks. In* Proceedings of the 10th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, *pages 129–136. IEEE.*

*Zacharias, S., Newe, T., OKeeffe, S., and Lewis, E. (2012). Identifying sources of interference in rssi traces of a single ieee 802.15.4 channel. In* Proceeding of the 8th International Conference on Wireless and Mobile Communications, *pages 408–414.*

*Zamalloa, M. and Krishnamachari, B. (2007). An analysis of unreliability and asymmetry in low-power wireless links.* Transactions on Sensor Networks, *3(2):7.*

*Zhang, L., Ferrero, R., Sanchez, E. R., and Rebaudengo, M. (2012). Performance analysis of reliable flooding in duty-cycle wireless sensor networks.* Transactions on Emerging Telecommunications Technologies.

*Zhang, Z., Zhou, H., and Gao, J. (2009). Scrutinizing performance of ad hoc routing protocols on wireless sensor networks. In* Proceedings of the 1st Asian Conference on Intelligent Information and Database Systems, *pages 459 –464.*

*Zhao, F. (2005). Challenge problems in sensornet research. Keynote at NSF NOSS PI meeting and Distinguished Lectures.*

*Zhao, J., Qiao, C., Sudhaakar, R., and Yoon, S. (2013). Improve efficiency and reliability in Single-Hop WSNs with transmit-only nodes.* IEEE Transactions on Parallel and Distributed Systems, *24(3):520 –534.*

*Zheng, J. and Lee, M. (2006). A comprehensive performance study of IEEE 802.15.4.* Sensor Network Operations, *pages 218–237.*

*Zheng, M., Yu, H., Liang, W., and Zhang, X. (2011). Optimal replicator factor control in wireless sensor networks.* Journal of Control Theory and Applications, *9(1):115–120.*

*Zhou, G., He, T., Krishnamurthy, S., and Stankovic, J. A. (2004). Impact of radio irregularity on wireless sensor networks. In* Proceedings of the 2nd international conference on Mobile systems, applications, and services, *pages 125–138.*

*Zhu, X. and Goldberg, A. (2009). Introduction to semi-supervised learning.* Synthesis lectures on artificial intelligence and machine learning, *3(1):1–130.*

*Zhuang, J., Jalloul, L., Novak, R., and Park, J. (2008). IEEE 802.16m evaluation methodology document (EMD).* IEEE C802. 16m-07/080r2, *802:16.*

*Zou, Y. and Chakrabarty, K. (2007). Redundancy Analysis and a Distributed Self-Organization Protocol for Fault-Tolerant Wireless Sensor Networks.* International Journal of Distributed Sensor Networks, *3(3):243–272.*

*Zowghi, D. and Gervasi, V. (2003). On the interplay between consistency, completeness, and correctness in requirements evolution.* Information and Software Technology, *45(14):993–1009.*