UNIVERSITÉ DU LUXEMBOURG

PhD-FSTC-36-2010
The Faculty of Sciences, Technology and Communication

DISSERTATION

Defense held on 17/12/2010 in Luxembourg

to obtain the degree of

DOCTEUR DE L'UNIVERSITÉ DU LUXEMBOURG

EN INFORMATIQUE

by

Apivadee PIYATUMRONG
Born on 29 April 1980 in Bangkok (Thailand)

ROBUST TOPOLOGY MANAGEMENT IN DT-MANETS: AN EFFICIENT TREE-BASED DECENTRALIZED AND MULTI-OBJECTIVE APPROACH

Dissertation defense committee

Dr Pascal BOUVRY, Dissertation supervisor
Professor, Université du Luxembourg

Dr Steffen ROTHKUGEL, Committee Chairman
Assistant Professor, Université du Luxembourg

Dr Frédéric GUINAND, Deputy Committee Chairman
Professor, Université du Havre

Dr Grégoire DANOY, Committee Member
Auxiliaire Scientifique, Université du Luxembourg

Dr Kittichai LAVANGNANANDA, Committee Member
Associate Professor, King Mongkut's University of Technology Thonburi

I would like to dedicate this thesis to my loving parents, Pichai Piyatumrong and Jarinee Angthong, and to my dear aunt, Vanida Kongcharoennivat.

Acknowledgements

I owe my deepest gratitude to my supervisor and both of my co-supervisors, Pascal Bouvry, Frédéric Guinand and Kittichai Lavangnananda, for their time, guidance and encouragement. Years of research and life during this PhD thesis gave a big impact to my scientific and cultural identity. I would like to show my gratitude to the members of *Teambouvry* for being good colleagues and cool friends. This thesis would not have been possible without their valuable comments, advices and supports. It is also a pleasure to thank many of my dear friends in Thailand, Luxembourg and other countries for their supports via all possible channels. Lastly, I offer my regards and blessings to all of those who supported me in any respect during the completion of this thesis.

Apivadee Piyatumrong (Sea)

Abstract

Delay-tolerant mobile ad hoc network (DT-MANETs) feature frequent and long duration partitioned MANETs. It is a challenged environment where end-to-end connectivity cannot always be obtained. Furthermore, communication in such network is heavily relying on collaboration between nodes since there is no central authority. Limited resources of communication nodes present another facet of problems in such network. Moreover, the communication is typically done using wireless technologies which are sharing among communication nodes. In order to provide a better quality of service in such environment, topology management technique is used to help controlling network topology. The aim is to manage network resource and to enhance efficiency of communication. In this work, we proposed to do it by constructing an efficient and robust tree-based topology. We model the environment using dynamic and partitioned graph. Under such circumstances, protocols must withstand topology and condition changing. In summary, doing topology management in DT-MANETs encounters the following issues: cooperation among nodes, limitation of resources of mobile node, sharing medium, dynamic and partitioned topology and unrealistic and unscalable approach of centralized algorithm in such environment.

This study focuses on managing tree-based topology in DT-MANETs. A set of active links is managed such that the deterioration of network is avoid and the quality of service in DT-MANETs is enhanced. Efficiency and robustness metrics are proposed accordingly. This work proposed to use different quality criteria based on communication node and edge for the selection topology. Both single- and multi-objectives tree-based topology are studied. Algorithms for managing tree-based topology are proposed according to different objectives. All proposed algorithms are purely decentralized, asynchronous algorithm and use one-hop information.

Contents

List of Figures xi

List of Tables xii

Definitions xiv

I Thesis 1

1 Introduction 2
1.1 Context . . . 3
1.2 Motivation . . . 4
1.3 Contributions . . . 4
1.4 Dissertation Outline . . . 5

2 Preliminary Concepts 7
2.1 Ad Hoc Networks and Modeling . . . 8
2.1.1 Overview of the Wireless Networking . . . 8
2.1.2 Ad Hoc Network . . . 10
2.1.2.1 Etymological Definition of Ad hoc . . . 10
2.1.2.2 Ad hoc, Wireless, Mobile, and Networks . . . 11
2.1.2.2.1 Ad hoc Networks . . . 11
2.1.2.2.2 Ad hoc Wireless Network . . . 12
2.1.2.2.3 Mobile Ad hoc Networks (MANETs) . . . 12
2.1.2.2.4 Self-Organization Networks . . . 13
2.1.2.3 MANET and its Related Networks . . . 14
2.1.3 Delay-Tolerant Mobile Ad Hoc Networks (DT-MANETs) . . . 17
2.1.3.1 Delay Tolerant and Disruption Tolerant Networks (DTNs) . 17
2.1.3.2 Definition, Characteristics and Challenges of DT-MANETs . 18
2.1.4 Modeling of an Ad hoc Network . . . 19
2.1.4.1 Graph Modeling . . . 20
2.1.4.2 Mobility Models . . . 21
2.2 Multi-Objectives Optimization Problem (MOOP) . . . 23
2.2.1 Introduction . . . 23
2.2.2 Related Definitions to MOOP . . . 24

2.2.3 Example Case and Multi-Objective Optimization Procedure 25
2.2.4 Optimization Algorithms 26
2.3 Summary 28

3 Topology management in Ad Hoc Networks **29**
3.1 Introduction 30
3.2 Relevant Topologies in MANETs 30
3.3 Topology Management 32
3.3.1 Motivation 32
3.3.2 Different Aspects of Topology Management 33
3.3.2.1 Physical Topology Management 33
3.3.2.2 Logical Topology Management (Virtual Backbone) 34
3.3.3 Mobility Issues and their Influences on Topology Management Algorithms 37
3.3.3.1 Centralized Algorithms 38
3.3.3.2 Distributed and Decentralized System 38
3.3.3.3 Desirable characteristics of algorithm for managing topology in DT-MANETs 39
3.4 Spanning Tree 40
3.4.1 Motivation 40
3.4.2 Constructing and Maintaining Spanning Forest in MANETs 40
3.4.3 Dynamicity Aware - Graph Relabeling System (DA-GRS) 41
3.4.4 Problems Relating to Spanning Tree 43
3.4.4.1 Minimum spanning tree (MST) problem 43
3.4.4.2 The multi-criteria MST problem 44
3.5 Robust and Efficient Topology Management in DT-MANETs 46
3.6 Summary 48

4 An Efficient and Robust Topology: Spanning Forest Approach **50**
4.1 Introduction 52
4.2 Efficient Spanning Forest for Topology Management 52
4.2.1 Motivation 52
4.2.2 Efficiency Metrics 53
4.2.2.1 spanningRatio() function 53
4.2.2.2 spanningSpeedRate() function 54
4.2.3 Utilizing and Applying Spanning Forest Algorithm of DA-GRS 54
4.2.3.1 Communicating Operation 54
4.2.3.2 Theoretical Performance 56
4.2.4 Efficient Spanning Forest by means of Tree Traversal Strategies 58
4.2.4.1 Introduction 58
4.2.4.2 State-of-the-Art 58
4.2.4.3 Proposition of a Solution: 'DFSmove' heuristic 60
4.3 Robustness Quality Criteria 62
4.3.1 Cooperative Enforcement Aspect 62
4.3.2 Energy Availability Aspect 63
4.3.3 Capacity Bandwidth Aspect 68
4.3.4 Discussion on other criteria 69

4.3.4.1 Signal Strength . . . 69
4.3.4.2 Age of edges . . . 70
4.4 Robustness Metrics . . . 71
4.4.1 nodeWeight() function . . . 71
4.4.2 isolateLowQualityNode() function . . . 72
4.4.3 pathWeight() function . . . 73
4.5 Robust Spanning Forest by Nodes and Edges' qualities . . . 75
4.5.1 Problem Description: Cooperative Enforcement Aspect . . . 75
4.5.2 Problem Description: Energy Availability Aspect . . . 76
4.5.3 Problem Description: Capacity Bandwidth Aspect . . . 77
4.6 Solutions for Robust Spanning Forest in DT-MANETs . . . 78
4.6.1 Introduction . . . 78
4.6.2 Construction and Maintenance Process . . . 79
4.6.2.1 G-NODE . . . 79
4.6.2.2 G-PATH . . . 79
4.6.3 Re-organization Process . . . 80
4.6.3.1 BREAK heuristic . . . 81
4.6.3.2 CHANCE heuristic . . . 81
4.7 Multi-objectives Approach for Managing Topology in DT-MANETs . . . 85
4.7.1 Problem Description . . . 86
4.7.2 Proposition of a solution: 'G-Node-Path' . . . 86
4.7.3 Optimum solution to single-objective function . . . 88
4.7.3.1 Reduction of $nodeWeight(\gamma)$ objective . . . 88
4.7.3.2 Reduction of $pathWeight(\gamma)$ objective . . . 90
4.7.4 Optimal solution to multi-objective function . . . 92
4.7.4.1 Enumerating the whole set of spanning trees: Exact Approach 92
4.7.4.2 Meta-Heuristics Approach . . . 93
4.8 Summary . . . 94

5 Experimentation and Validation Methodologies **98**
5.1 Experimental Setup . . . 99
5.1.1 Used Simulators . . . 99
5.1.2 Analysis of Used Mobility models . . . 100
5.1.3 Assumptions . . . 107
5.1.3.1 datagram used in this work . . . 107
5.1.3.2 Communication Assumptions in GraphStream . . . 107
5.2 Results and Analysis of the Proposal . . . 111
5.2.1 Tree Traversal Strategies . . . 111
5.2.1.1 Preliminary . . . 111
5.2.1.2 Tree Traversal Strategies over DA-GRS reference algorithm 111
5.2.1.3 Tree Traversal Strategies over the Robustness of Trusted Tree Topology by G-NODE . . . 114
5.2.2 Robust topology utilizing Cooperative Enforcement Approach . . . 118
5.2.3 Robust topology utilizing Energy Availability Approach . . . 121
5.2.4 Robust topology utilizing Capacity Bandwidth Approach . . . 124

5.2.5 Reorganization of topology by BREAK heuristic 127
5.2.6 Reorganization of topology by CHANCE heuristic 130
5.2.7 Efficient and Robust Topology in Multi-Objective Optimization Approach 134
5.2.7.1 Experimentation methodology and Implementation note . . . 134
5.2.7.2 Hypervolume results 137
5.2.7.3 Robustness Metrics Results 138
5.3 Summary 139

6 Conclusions and Perspectives 140
6.1 Conclusion 141
6.1.1 The Framework 141
6.1.2 Robust Quality Criteria 141
6.1.3 Solution Algorithms and Heuristics 142
6.1.3.1 Token traversal strategy 142
6.1.3.2 Construction and maintenance process 142
6.1.3.3 Re-organization process 143
6.1.4 Evaluation Methodology 143
6.2 Perspective 144

II Appendix 146

A Results in Graph 147
A.1 Robust topology utilizing Cooperative Enforcement Approach 148
A.2 Robust topology utilizing Energy Availability Approach 152
A.3 Robustness and Efficiency Results for CHANCE heuristic 156
A.4 Overhead Messages for CHANCE heuristic 160
A.5 Hypervolume Result 168
A.6 Robustness Metric Result for Multi-objective problem 170

References 172

Publications 184

Index 184

List of Figures

2.1 Internet and wireless network 10
2.2 Example of two fix stations composing an ad hoc network using a crossover cable 11
2.3 Example of three fixed stations equipped with multiple wireless communicating interfaces forming ad hoc networks using different enabling technologies; Wi-Fi and Bluetooth 12
2.4 Relationship between DT-MANETs, MANETs, WSNs, WMNs, and VANETs 19
2.5 Hypothetical trade-off solutions for acquiring a mobile phone 25
2.6 Multi-objective optimization procedure 26
2.7 Taxonomy of Optimization Techniques [1] 27

3.1 Examples of Topology in MANETs 31
3.2 Taxonomy of logical topology structure or virtual backbone 35
3.3 DA-GRS rules for creating and maintaining spanning forest topologies with our mapping process name 42

4.1 An example scenario for illustrating the proposed efficiency metrics 54
4.2 Heuristics used in DA-GRS as token traversal strategy and Merging Process Strategy 55
4.3 Message sequence diagram for merging trees 56
4.4 Message sequence diagram for traversing the token 56
4.5 Example topology used for complexity explanation 57
4.6 An example of no convergence case using DA-GRS reference algorithm 57
4.7 Relation between trust and reputation 63
4.8 General Discharge Curve Characteristic 65
4.9 discharge characteristic of different cell chemistry [2],[3] 66
4.10 Screenshot examples of battery applications in mobile phones 68
4.11 An example scenario for Illustrating the details of *nodeWeight()* and *isolateLowQualityNode()* cost functions 72
4.12 An example scenario for Illustrating the details of both efficiency and robustness metrics 72
4.13 An example scenario for Illustrating the details of robustness metric of pathWeight() function 74
4.14 An example scenario for Illustrating the details of robustness metrics for percentage of battery level case 76

4.15 Framework proposed by this study including Robust Quality Criteria, Token Traversal Strategy and Metrics for Efficiency and Robustness 78
4.16 An example illustrating BREAK heuristic 82
4.17 An example illustrating CHANCE heuristic in G-NODE (concerning node's quality) 83
4.18 Framework proposed by this study including construction, maintenance and re-organization process 85
4.19 Relationship between G-NODE, G-PATH and G-Node-Path 87
4.20 An example illustrates the equivalent of robust spanning forest topology problem to maximum spanning tree 89
4.21 Example graph and spanning tree for reduction pathWeight() function proof 90
4.22 example of max pathWeight() spanning tree 91
4.23 Proposed Framework for Topology Management in DT-MANETs based on Spanning Forest 95
4.24 Purely decentralized and one-hop information used heuristics proposed by this work 96

5.1 An example of Shopping Mall Mobility Model with 240 nodes 100
5.2 An Example of Highway Mobility Model with 80 nodes 101
5.3 Number of Edges and Connected Component of each configurations from both Mobility Model according to Table 5.1 and 5.2 103
5.4 An example of box plot 104
5.5 Box plot showing edge's age characteristic of each configurations of Shopping Mall Mobility Model according to Table 5.1 105
5.6 Box plot showing edge's age characteristic of each configurations of Highway Mobility Model according to Table 5.2 106
5.7 Packet Structure used for G-NODE and G-PATH heuristics 107
5.8 Comparison of spanningSpeedRate() measuring among all studied algorithms in 'Shopping Mall' mobility model 112
5.9 Comparison of spanningSpeedRate() measuring among all studied algorithms in 'highway' mobility models 113
5.10 The result based on 600 runs comparing the number of iteration used in obtaining trees using different strategies (RANDOMmove, TABUmove and DFSmove) 114
5.11 Comparison of spanningRatio(), spanningSpeedRate(), nodeWeight() and isolatingLowQualityNode() measuring among all studied algorithms in Shopping Mall Mobility Model 115
5.12 Comparison of spanningRatio(), spanningSpeedRate(), nodeWeight() and isolatingLowQualityNode() measuring among all studied algorithms in Highway Mobility Model 116
5.13 Example of Contrasting between spanningRatio() and isolatingLowQualityNode() 117
5.14 Result of G-NODE utilizing trust value compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model 119
5.15 Result of G-NODE utilizing trust value compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model 120

5.16 Result of G-NODE utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model 122
5.17 Result of G-NODE utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model 123
5.18 Result of G-PATH utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model 125
5.19 Result of G-PATH utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model 126
5.20 Comparison of nodeWeight() and isolateLowQualityNode() functions measuring among all studied algorithms 128
5.21 G-NODE CHANCE on Shopping Mall Configuration 4 130
5.22 G-NODE CHANCE on Highway Configuration 4 131
5.23 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway and Shopping Mall Configuration 4 133
5.24 Examples of snapshots of the simulation 134
5.25 Examples of communication graph at different time step t with emphasizing of connected components in red circles 134
5.26 Finding Pareto front of a graph G(t) 135
5.27 Multi-objective search space 136
5.28 The preference trade off value, ρ, and the experiment methodology 137
5.29 Comparison between hypervolume of NSGA-II, DA-GRS and G-Node-Path over simulation time 137
5.30 Comparison between path quality and node quality of five different preference value (ρ) over simulation time 138

A.1 Cooperative Enforcement Approach: G-NODE on Highway Configuration 1 . 148
A.2 Cooperative Enforcement Approach: G-NODE on Highway Configuration 2 . 148
A.3 Cooperative Enforcement Approach: G-NODE on Highway Configuration 3 . 149
A.4 Cooperative Enforcement Approach: G-NODE on Highway Configuration 4 . 149
A.5 Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 1 150
A.6 Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 2 150
A.7 Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 3 151
A.8 Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 4 151
A.9 Energy Availability Approach: G-NODE on Highway Configuration 1 152
A.10 Energy Availability Approach: G-NODE on Highway Configuration 2 152
A.11 Energy Availability Approach: G-NODE on Highway Configuration 3 153
A.12 Energy Availability Approach: G-NODE on Highway Configuration 4 153
A.13 Energy Availability Approach: G-NODE on Shopping Mall Configuration 1 . 154
A.14 Energy Availability Approach: G-NODE on Shopping Mall Configuration 2 . 154
A.15 Energy Availability Approach: G-NODE on Shopping Mall Configuration 3 . 155
A.16 Energy Availability Approach: G-NODE on Shopping Mall Configuration 4 . 155
A.17 G-NODE CHANCE on Highway Configuration 1 156

A.18 G-NODE CHANCE on Highway Configuration 2 156
A.19 G-NODE CHANCE on Highway Configuration 3 157
A.20 G-NODE CHANCE on Highway Configuration 4 157
A.21 G-NODE CHANCE on Shopping Mall Configuration 1 158
A.22 G-NODE CHANCE on Shopping Mall Configuration 2 158
A.23 G-NODE CHANCE on Shopping Mall Configuration 3 159
A.24 G-NODE CHANCE on Shopping Mall Configuration 4 159
A.25 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 1 160
A.26 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 2 161
A.27 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 3 162
A.28 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 4 163
A.29 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 1 164
A.30 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 2 165
A.31 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 3 166
A.32 Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 4 167
A.33 Hypervolume: G-Node-Path, DA-GRS and Pareto front on Highway Configuration 1 168
A.34 Hypervolume: G-Node-Path, DA-GRS and Pareto front on Highway Configuration 2 168
A.35 Hypervolume: G-Node-Path, DA-GRS and Pareto front on shopping mall Configuration 1 169
A.36 Hypervolume: G-Node-Path, DA-GRS and Pareto front on shopping mall Configuration 2 169
A.37 G-Node-Path at different preference value on Highway Configuration 1 170
A.38 G-Node-Path at different preference value on Highway Configuration 2 170
A.39 G-Node-Path at different preference value on shopping mall Configuration 1 . 171
A.40 G-Node-Path at different preference value on shopping mall Configuration 2 . 171

List of Tables

2.1 Example of specification of today's mobile phone 9

4.1 General specification of different battery chemistry type 66
4.2 Example of specification of mobile nodes' batteries 67
4.3 Summary heuristic used by DA-GRS and G-NODE at each activity concerning spanning forest construction and maintenance using token management strategy 80
4.4 Result of Enumerating Whole Set of Spanning Trees using Different Non-Complete Graph and Size of Vertices 93
4.5 Enumerating all Spanning Trees Results from Complete graphs 93

5.1 Parameters used in the experiments for Shopping Mall mobility model 101
5.2 Parameters used in the experiments for Highway mobility model 102
5.3 Size of each packet type used in this work 108
5.4 Calculation of capacity of wireless network following Gupta's finding 109
5.5 Parameters used in the experiments 111
5.6 The percentile of the distance from DFSmove to the other strategies 112
5.7 Multi-objective multi-constraint study 113
5.8 Parameterization used in Madhoc 127
5.9 Overhead comparison among 'BREAK' heuristic at different *T_break* in G-NODE on 'highway' model 129
5.10 Overhead comparison among 'BREAK' heuristic at different *T_break* in G-NODE on 'shopping mall' model 129

List of definitions

1 Ad Hoc - Longman Dictionary 10
2 Ad Hoc - Cambridge Advanced Learner's Dictionary 10
3 Ad Hoc - Merriam-Webster's Online Dictionary 10
4 Ad Hoc Networks - Charles E. Perkins 11
5 Ad Hoc Networks - Our Definition 11
6 1-hop neighbor 11
7 Ad hoc Wireless Network - C. Murthy 12
8 Characteristics of Mobile Ad hoc Networks (MANETs) – IETF RFC2501 13
9 Self-Organization 14
10 Delay-tolerant mobile ad hoc networks 18
11 Model 20
12 Communication Graph 20
13 One-hop Neighbors – in Communication Graph 20
14 Dynamic Communication Graphs 21
15 Connected Component of a Dynamic Communication Graph 21
16 Mobility Models 21
17 Optimize – Cambridge Advanced Learner's Dictionary 23
18 Instances of an optimization problem 23
19 Optimization problem 23
20 Multi-Objective Optimization Problem 24
21 Pareto Dominance 24
22 Pareto Optimality 24
23 Pareto Optimal Set 24
24 Pareto Front 24
25 Topology – relevant to networking 30
26 Logical (virtual) topology 30
27 Connected Graph 31
28 Subgraph 31
29 Complete Graph 32
30 Topology Management (TM) 32
31 Topology Management Algorithm 33
32 Dominating Set (DS) 35

33 Connected Dominating Set (CDS) 36
34 Clustering 36
35 Independent set 36
36 Spanning Tree 40
37 Spanning Forest 41
38 Minimum Spanning Tree (MST) 43
39 mc-MST problem 45
40 Robust - Cambridge Advanced Learner's Dictionary 46
41 Efficient - Cambridge Advanced Learner's Dictionary 46
42 Conflicting objectives 48
43 Trust 62
44 Reputation - Cambridge Advanced Learner's Dictionary 62
45 Capacity Bandwidth 69
46 Path 69
47 The Matrix-Tree Theorem 92

Part I

Thesis

Chapter 1

Introduction

Contents

1.1 Context 3
1.2 Motivation 4
1.3 Contributions 4
1.4 Dissertation Outline 5

1.1 Context

Computer network is one of the most important breakthrough in the twentieth century. The growth of Internet subscriber increase dramatically over past twenty years. This is even more prominent with the wireless network era, which has its based on cellular networks. Wireless communication is becoming more and more mandatory in households, business and public sector. Many big cities around the world now are providing free-wireless Internet access at some particular areas. There has been an attempt to extend the Internet coverage into a hard-to-provide service area or the place where the implementation is too expensive by using ad hoc networks. This opens up a lot of possibilities and applications (i.e., inexpensive advertisement, communication in emergency situation, combat and rescue). Ad hoc draws a great deal of attentions from many researchers and among different research fields too.

To date, the focuses on ad hoc network have been on the core technology itself (i.e., wireless channel modulation, hardware characteristics, encoding scheme, etc.) and their applications. While application inspires technology, the technology also enables the applications. A very popular application for ad hoc technology is for portable console gamer. A meeting for playing the PlayStation Portable (PSP) or the Nintendo DS series in a coffee shop or fast food restaurants become a common activity in Japan and other places in Asia. Files and photos can be passing from one mobile phone to another mobile phone. Ad hoc network enabling these activities via WiFi and Bluetooth technology. These examples present wireless ad hoc network in a static scenario. Nowadays, lifestyle of urban people have changed rapidly. The desire to keep in touch any time and every where is taken for granted. Accordingly, telecom providers must increase the signal coverage area by installing more infrastructures. Unfortunately, such solutions are too costly comparing to gained benefit. This issue is very more apparent in rural area. There are several projects trying to provide a low cost Internet and telephone access for such area without fully installed infrastructures. It seems ad hoc network is one of the solutions. Most of ad hoc solutions, such as the last mile broadband Internet access, utilize a so called 'hybrid' solution (which is a combination between some infrastructure nodes and a few 'on the fly' ad hoc nodes with some constraints, such as having low mobility). There are many technical issues prohibiting the full utilization of purely ad hoc networks which researchers are trying to overcome.

Major research works regarding ad hoc networks basically follow what exist in wired or classical network such as routing protocol, security issue, encoding and decoding issue, topology management, etc. However, research works in ad hoc network is much more complex than those found in classical network due to several factors. First, ad hoc networks typically use wireless technologies which are shared medium. This means one can neither guarantee the bandwidth usage over a communication period, nor the success of communication. Second, there are limitations of computation and memory resources in each mobile node. Third, a mobile node in an ad hoc network can change its location, any protocol operating in ad hoc networks must have a high level of flexibility, in order to adapt to uncertainty of location and topology of nodes.

Visioning beyonds such hybrid network, this dissertation focuses on purely ad hoc, dynamic, and disrupted network, so called delay-tolerant mobile ad hoc networks (DT-MANETs). The primitive characteristics of such networks, which are the uncertainty of its dynamicity and the partitioning of the network, present the broad context of this study. Topology management techniques are well-known techniques which are applied to overcome some problems existing in network. Narrowing the context down, this research work aims to enhance the quality of communication and services of DT-MANETs by means of topology management. Contributions and novel of the work lies in its ability to manage change and uncertainty occurring from the nature of DT-MANETs in decentralized manner by having efficient topology management and robust spanning forest topology at different level of constraints.

1.2 Motivation

Ad hoc network gained the vast majority of interest in the past 10-15 years. Many applications of ad hoc network are functioning worldwide. Many conferences and research works are being carried out and brought in discussions at current moment. We can expect that these research works become more an more challenging in years to come. The primitive concern of communication is quality of services (QoS). Due to the lacking of central authority in ad hoc network, provisioning and managing resources are difficult. Topology management has its aim to handle or control the network topology in order to provide better control over network resources and to increase the efficiency of communication. The real challenge lies in inventing and implementing decentralized, asynchronous algorithm which utilizes only local knowledge within dynamic ad hoc network scheme.

Many topology management based on spanning tree topology have been proposed to tackle various problems under mobile ad hoc network context. They are different both in terms of assumptions and operational process. Some of them proposed centralized algorithm, while many proposed distributed algorithm but sharing global information. Hence, their applications on real devices is less than realistic and they do not scale well. Furthermore, mobility assumptions are limited from static to slow mobility. Very few works are found to provide solutions for a purely decentralized, asynchronous, where only local knowledge algorithm is used.

Ad hoc networks encounter wide variety of obstacles, such as signal interference, mobility of nodes, limited power, density of nodes, etc. Under such a complex scenario, multiple criteria should be taken into account. Hence, the problem should be considered as a multi-objective problems. However, few works provide solutions to a conflicting criteria problem of ad hoc networks.

This research focuses in particular on constructing robust spanning forest topology and managing it efficiently in a dynamic and disrupted ad hoc network, DT-MANETs. All contributions comply with the desired characteristics of algorithm which is lacked in previous work. In this study, structure quality is concerned at three different perspectives. First is at the overall communication graph level which is related to communication efficiency. Second focuses at the quality or robustness of each constructed tree topology. Third is at the qualities criteria found in nodes and edges. Problems in both single-objective and multi-objective approaches are analyzed and solutions by means of algorithms and heuristics are implemented. Altogether, three distributed algorithms and three distributed heuristics are introduced.

1.3 Contributions

Contributions to the body of knowledge in this study can be summarized as follows:

- Studying of the robustness and efficiency issues for DT-MANETs, in particular for tree-based management structures. This includes the selection of suitable quality criteria in entity like communication node and communication edge.
- A token traversal strategy for tree-based management structure that can cover the network larger and faster than the strategy used in previous works.
- A set of purely decentralized, asynchronous and using local knowledge algorithms for tree-based management structure are proposed by utilizing quality criteria of nodes and edges.
- Re-organization heuristics for improving tree-based management structure that operates in purely decentralized and using only local knowledge are introduced.
- A decentralized, asynchronous, and using only local knowledge algorithm for providing solutions according to a specific set of preferences for multi-objective spanning tree topology is proposed.
- A framework including all plug-in algorithms and heuristics for tree-based management structure is proposed. Experimental and validation modules for both single-objective and multi-objectives approach are also analyzed in detail.

1.4 Dissertation Outline

The rest of the dissertation is organized as follows:

Chapter 2 describes the definitions and examples of ad hoc network and its related networks. The conceptual and definition of DT-MANETs can be found in this Chapter. The Chapter also gives a discussion on its modeling which is based on graph theory. Furthermore, the basic of multi-objectives optimization problem is explained with insight examples.

Chapter 3 generalizes the idea of topology management in ad hoc networks. This Chapter explains the approaches found in literature for managing topology in ad hoc networks. It discusses the effects of mobility issue to topology management and suggests the suitable characteristics of the algorithm for managing topology in DT-MANETs. Furthermore, robust and efficient topology management are discussed briefly. The Chapter also reviews the literature on topology management as well as literature of spanning tree and its construction and maintenance in ad hoc networks.

Chapter 4 is the main chapter which contains the description and discussion of all contributions. It comprises different sections explaining the motivation, idea, cost functions (metrics) and the proposed solution algorithms or heuristics. It provides the discussion on both efficient and robust spanning forest. Efficient metrics and solution to obtain a better efficiency spanning forest are discussed. Robustness quality criteria and their characteristics proposed in this study are elaborated in detail. Also, the multi-objectives approach for managing topology in DT-MANETs is presented along with the problem description, the solution algorithm and the proof of optimality for single-objective problem. The methodology to obtain the Pareto-front for multi-objective function is described and examples are given.

Chapter 5 illustrates the experimentation and validation methodologies. Since, the experimentation of this study is simulation based, the rationale and the simulators used in this study are explained here. Results and analysis of all proposed algorithms and heuristics are provided.

Chapter 6 concludes the dissertation with the summary of all contributions and novelty of this study. It also puts the study into perspectives of the DT-MANETs community. Future work and directions for further investigation are also suggested.

Chapter 2

Preliminary Concepts

Contents

2.1 Ad Hoc Networks and Modeling **8**
- 2.1.1 Overview of the Wireless Networking 8
- 2.1.2 Ad Hoc Network 10
  - 2.1.2.1 Etymological Definition of Ad hoc 10
  - 2.1.2.2 Ad hoc, Wireless, Mobile, and Networks 11
    - 2.1.2.2.1 Ad hoc Networks 11
    - 2.1.2.2.2 Ad hoc Wireless Network 12
    - 2.1.2.2.3 Mobile Ad hoc Networks (MANETs) 12
    - 2.1.2.2.4 Self-Organization Networks 13
  - 2.1.2.3 MANET and its Related Networks 14
- 2.1.3 Delay-Tolerant Mobile Ad Hoc Networks (DT-MANETs) 17
  - 2.1.3.1 Delay Tolerant and Disruption Tolerant Networks (DTNs) 17
  - 2.1.3.2 Definition, Characteristics and Challenges of DT-MANETs 18
- 2.1.4 Modeling of an Ad hoc Network 19
  - 2.1.4.1 Graph Modeling 20
  - 2.1.4.2 Mobility Models 21

2.2 Multi-Objectives Optimization Problem (MOOP) **23**
- 2.2.1 Introduction 23
- 2.2.2 Related Definitions to MOOP 24
- 2.2.3 Example Case and Multi-Objective Optimization Procedure 25
- 2.2.4 Optimization Algorithms 26

2.3 Summary **28**

2.1 Ad Hoc Networks and Modeling

This chapter discusses the overview of classical networking to ad hoc networking and the relevance types of ad hoc networks. It describes ad hoc network model from a basic to a more sophisticate. Furthermore, the modeling basis of such networks for evaluation purpose are given briefly.

2.1.1 Overview of the Wireless Networking

Before going into the specifics of ad hoc networks, a classical computer networks is introduced. Generally, a classical computer network is a collection of computers and devices connected via a set of communication infrastructures. These infrastructures are pre-established in order to provide the communication both using wired physical media (i.e. coaxial cable, optic fiber, etc.) and wireless physical media (i.e. satellite, microwave, radio). With the differences of used medium, there exists a wide range of ways to utilize and/or enhance the usability of the existing network.

The basic idea of computer network is to share resources and to communicate for collaborative work. It is not surprising why computer networks become very essential for day-to-day life and are used by a wide range of applications, such as e-mail systems, ticket reservation system, video conferencing, Voice over IP (VoIP), learning on-demand system, social media and social networking systems. In household, people connect to the Internet via modem and may create their own local area network (LAN) or wireless local area network (WLANs) for all members of the family for sharing Internet or playing games. Likewise, in company, networking becomes a part of their working life by ordering printer to print report, using LAN or wide area network (WAN) for making order to stock or checking the back office system. Sales and marketing person use more and more mobile devices checking into the database of the company to get realtime information via cellular networks. Restaurants are applying the ordering system and credit card authentication system wirelessly. From those examples, there are facts that the computer networks has been heavily used in the last decade. Incorporating with the growth number and capability of portable devices (i.e. smart phone, PDA and laptop), one cannot deny the importance of wireless communication networks.

Wireless communications is one of the fastest growing industries in the world. The major portion of this fact has been due to cellular networks. Cellular networks provide two-way simultaneous voice communications. A fixed base station (BS) serves all mobile phones in its coverage area, called a cell. The cellular networks have been developing the capacity and designed services through time. From the first generation (1G) cellular networks that used analog signal technology, we already experienced 2.5G (GPRS) and 2.75G (EDGE) during year 2000. Since then the technologies provide data and multimedia services through mobile devices. At present, we are experiencing the third-generation (3G) which provides the maximum peak bit rate supported at about 2 Mbps (according to standard IMT-2000). Furthermore, the implementation of 3G transitional (marked by 3.9G or Long Term Evolution, LTE) which has been launched in two capitals of Europe (Stockholm and Oslo) in December 2009 [4]. The LTE is a step toward the 4th generation (4G) of radio technologies but it is still a 3.9G technology since it does not fully comply with the IMT Advanced or 4G requirement. LTE is specified by Third-Generation Partnership Project (3GPP) to provide downlink peak rates around 75 Mbps. The current standardization of LTE allows for speeds at 300Mbps [5].

Although, the latest mobile device such as 'HTC Desire' [6] can operate up to 7.2 Mbps download speed and up to 2 Mbps upload speed, the much more powerful mobile phone can be expected to match the 4G technology very soon.

In this era, mobile devices have higher capability and functionality (e.g. power, memory, processor). Table 2.1 shows us some of current smartphone's specifications [7], [8], [9], [10], and [11] (Please note that the 'Talk time' provided in Table 2.1 are all theoretical value which provided by each mobile phone's company). Thus, a vast majority of developers and practitioners are focusing on developing various applications on and enhancing the capability of smartphone (iphone, BlackBerry, HTC etc.).

Table 2.1: Example of specification of today's mobile phone

Smartphone Models	CPU Processing Speed	RAM	Memory Storage	Power (Talk time)
iPhone 3Gs	600 MHz	256 MB	16/32 GB flash drive	300 hrs. 300 hrs.
HTC desire	1 GHz	576 MB	up to 32 GB $microSD^{TM}$	360 hrs.
BlackBerry Bold 9700	624 MHz	128 MB	$microSD^{TM}$ expansion slot	408 hrs.
DROID by Motorola	550 MHz	-	16 GB $microSD^{TM}$ + up to 32 GB expansion slot	270 hrs.

Apart from Cellular Mobile Network, wireless local area networks (WLANs) are another type of wireless networks that are basis of today life. Nowadays, many households, airports and other public places all over the world operate Wireless LAN. According to the 802.11 LAN standard, operation of wireless LAN (WLAN) can be under two configurations, with base station and without a base station. In the former, all communications must go through the base station, called as an access point (AP). The AP is pre-established in the location for providing WLAN such as in university, waiting room in hospital, airport and coffee shop. In the latter, the computers and communicating devices communicate by broadcasting information to others locally. This mode is called ad hoc networking and is discussed more in details in the coming Section 2.1.2.

Figure 2.1 illustrates and summarizes those quick overview that we discussed so far. On the left of the figure, a WLAN is presented with an existing of AP. Notebook and mobile materials equipped with wireless communicating devices can connect through Internet and to each other through this AP. This WLAN is also connected to Ethernet LAN and Intranet of an organization. On the other hand, at the right-most of the figure, the cellular network is presented. Mobile users connect through a based station which has a pre-established wired link connected with a Mobile Switching Center. For some nodes that cannot get directly through the existing infrastructure but stay in range of some devices connecting with a base station (bottom right), they use infrastructureless mode, wireless ad hoc network, to communicate through cellular network. The mixing of both infrastructure and infrastructureless is called hybrid wireless network. These hybrid architectures (which combine the benefits of cellular and ad hoc wireless networks) are expected to improve the capacity of the system significantly. In the bottom middle of the figure, a wireless ad hoc network has been shown as a set of devices communicate to each other without a single pre-existed infrastructure.

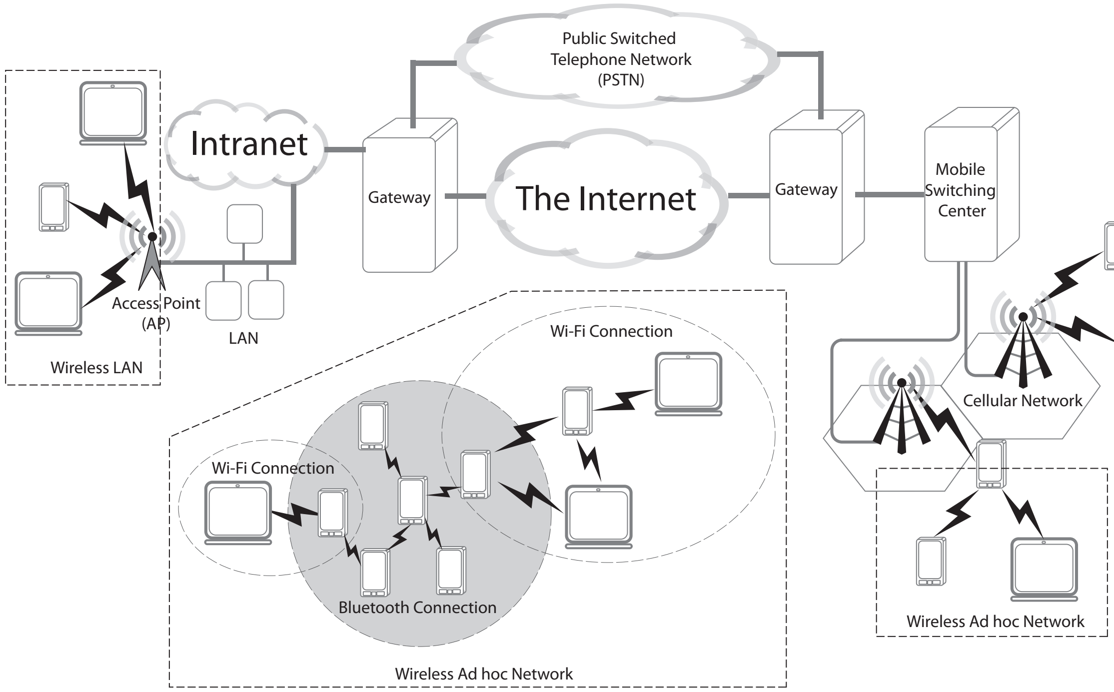


Figure 2.1: Internet and wireless network

2.1.2 Ad Hoc Network

2.1.2.1 Etymological Definition of Ad hoc

First, let us start our discussion with definitions of the word 'Ad Hoc' from various few well known English dictionaries.

Definition 1 (Ad Hoc - Longman Dictionary).
not planned, but arranged or done only when necessary

Definition 2 (Ad Hoc - Cambridge Advanced Learner's Dictionary).
made or happening only for a particular purpose or need, not planned before it happens

Definition 3 (Ad Hoc - Merriam-Webster's Online Dictionary).
formed or used for specific or immediate problems or needs (ad hoc solutions)

Furthermore, the word 'ad hoc' can be traced from its Neo-Latin origin, created between 16th and 20th and employed in scholarly and scientific publications. The direct translation of the word ad hoc means 'for this'. The term is generally used to imply that something is intended for a particular situation or purpose.

2.1.2.2 Ad hoc, Wireless, Mobile, and Networks

2.1.2.2.1 Ad hoc Networks

Definition 4 (Ad Hoc Networks - Charles E. Perkins).
An ad hoc network is one that comes together as needed, not necessarily with any assistance from the existing Internet infrastructure. [12]

Definition 5 (Ad Hoc Networks - Our Definition).
An ad hoc network is a spontaneous, an infrastructure-less and an autonomous network. This means the network has no established infrastructure or centralized administration, such as router or Dynamic Host Configuration Protocol (DHCP). As a result of no-infrastructure, the network has to operate in distributed peer-to-peer mode, acts as an independent router and generates independent data.

In fact, such a feature is applicable with both stationary (fixed) and mobile nodes, using wired or wireless technologies. Under this section let us first discuss only fixed stations and wired technology. The basic example for an ad hoc networks is two computers connecting via crossover cable, shown in Figure 2.2, and become a 1-hop neighbor of each other.

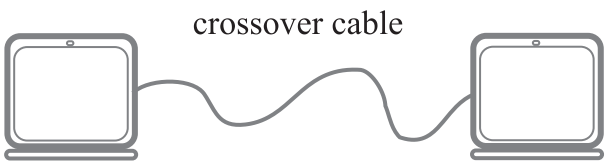


Figure 2.2: Example of two fix stations composing an ad hoc network using a crossover cable

Definition 6 (1-hop neighbor).
A node X is a 1-hop neighbor of a node Y if a communication interface of node X is heard by a communication interface of node Y. [13].

In networking, the hop count represents the total number of devices that a given piece of data (packet) passes through. Normally, such counting is used to measure the cost and time being used for communicating between the source and the destination nodes (e.g. using traceroute). However, in a flat and purely ad hoc network, each node acts both as router and normal client. The definition 6 regards each nodes as a router and determine direct (1-hop) connectivity to neighboring nodes (routers). The above example of connectivity between two computers, thus, becomes one-hop communication.

2.1.2.2.2 Ad hoc Wireless Network

Definition 7 (Ad hoc Wireless Network - C. Murthy)**.**
An ad hoc wireless network is an autonomous system of nodes connected through wireless links. It does not have any fixed infrastructure (such as base stations in cellular networks). The machines in the network coordinate among themselves for communication. Hence, each node in the network, apart from being a source or destination, is also expected to route packets for other nodes in the network. [14]

Typical fixed infrastructures in wireless network are either base station (BS) in cellular networks or access point (AP) in IEEE802.11 WLAN. Without those pre-established devices, wireless nodes can communicate in peer-to-peer mode using broadcasting as a primitive operation. As an addition to ad hoc network, an ad hoc wireless network has a different feature in variation of wireless link and node capabilities. Each node maybe equipped with one or more interfaces that have varying transmission/receiving capabilities and operate across different technologies (i.e. Wi-Fi, Bluetooth, etc.). This heterogeneity in node radio capabilities can result in possibly asymmetric links.

In fact, the big issue that makes ad hoc wireless networking popular is the feature of multi-hop relaying. A set of middle nodes between source and destination nodes relay the message on behalf of source node to destination and thus act as routers. The routing decisions made by those middle nodes has been done in a decentralized manner, which, is indeed, totally different from a centralized routing mechanism processing by base stations in classical/infrastructure networks. Figure 2.3 depicts an example of ad hoc wireless networks where all nodes are fixed stations equipped with wireless communicating devices and formed a network together.

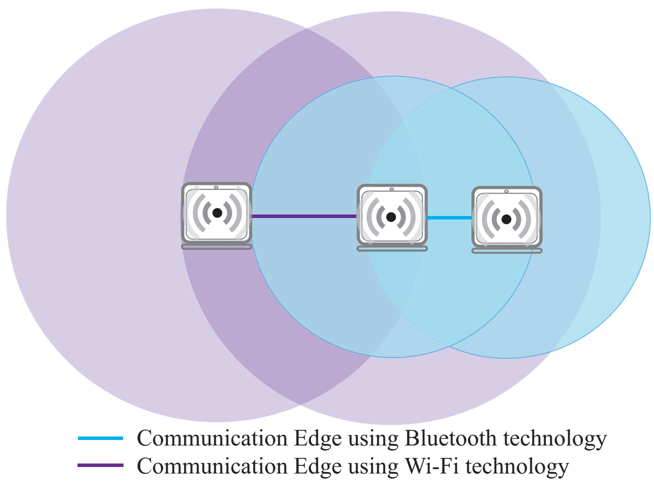


Figure 2.3: Example of three fixed stations equipped with multiple wireless communicating interfaces forming ad hoc networks using different enabling technologies; Wi-Fi and Bluetooth

2.1.2.2.3 Mobile Ad hoc Networks (MANETs)

The terminology 'MANET' has been used officially by IETF RFC2501 [15], Mobile Ad hoc

Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations, for describing the technology since January 1999.

"The technology of Mobile Ad hoc Networking is somewhat synonymous with Mobile Packet Radio Networking, Mobile Mesh Networking, and Mobile, Multihop, Wireless Networking (perhaps the most accurate term).", [15]. RFC2501 gave an extensive characteristics of MANETs which can be summarized as follows:

Definition 8 (Characteristics of Mobile Ad hoc Networks (MANETs) – IETF RFC2501)**.**
A MANET consists of mobile platforms, simply referred to nodes, which are free to move about arbitrarily. A MANET is an autonomous system of mobile nodes. The system may operate in isolation, or may have gateways to and interface with a fixed network. MANET nodes are equipped with wireless transmitters and receivers using antennas which may be omnidirectional (broadcast), highly directional (point-to-point), possibly steerable, or some combination thereof. At a given point in time, depending on the nodes' positions and their transmitter and receiver coverage patterns, transmission power levels and co-channel interference levels, a wireless connectivity in the form of a random, multihop graph or 'ad hoc' network exists between the nodes. This ad hoc topology may change with time as the nodes move or adjust their transmission and reception parameters.

To the context of this work, in short MANET is a spontaneous, infrastructure-less, autonomous, wireless and composing of mobile nodes. A MANET includes all properties of being ad hoc and ad hoc wireless networks together with the special ability to move of stations. As an example, suppose some bio-chemical weapons were dispersed in some area. Instead of sending chemical engineers and polices into the poisonous area, it would be better to distribute sensors prompted to work in ad hoc mode into the area. After its situated, the sensor could form an ad hoc wireless network and cooperate to gather the desired information about the poison back to the emergency center. Then, the center can send in a set of robot agent (mobile stations) to clean out the weapons. These robots may cooperate between each other and also the pre-established sensors (fixed stations) using ad hoc network. From the example above, an ad hoc network does not require any infrastructure. It uses *wireless technologies*, while nodes in mobile ad hoc networks can be fixed nodes and/or mobile nodes.

Many challenges occur when a station of a network can move. The mobility of mobile nodes makes self-configuring and the decentralized routing process of the network much harder than a fix ad hoc network. Dealing with very mobile nodes is one of the challenge which motivates this research to focus on dynamic networks at different mobility types. It can be summarized that uncertainty of nodes' position has the most effect to the self-organization. Thus, we give an explanation briefly on this important feature as follows.

2.1.2.2.4 Self-Organization Networks

In IETF RFC2501, it highlights that mobile nodes need to behave like an autonomous system. It was IBM who formally introduced the term autonomic system for a system that is less dependent on human intervention and able to cope by itself with the complexity and heterogeneity of its life cycle [16]. In [17], the notion of self-organization as defined by IBM is instantiated for the mobile ad hoc network context.

Definition 9 (Self-Organization).
An ad hoc network is a self-organization system if the network organizes and maintains the network by itself, such as neighbor discovery, topology organization, and topology reorganization. These processes also can be found in name of Self-Configuring and Self-Healing.

A self-organization system must configure and reconfigure itself under varying (and in the future, even unpredictable) conditions. System configuration or 'setup' must occur automatically as well as dynamic adjustments so that the best configuration is used to handle changing environments [16]. In the context of this work, a self-organized wireless network should have some discovery mechanisms implemented. Nodes should know their environment, in particular they should be able to detect their neighbors and when one of them leaves the neighborhood, the node should have some mechanisms to deduce that information. Furthermore, during the maintenance period, the self-organized network must be able to recover from routine events of MANETs such as link failures, node shutting down, etc. The system must have a way to reconfiguring the system to keep functioning smoothly based on the current situation and information. In summary, the aspect of self-organization is crucial for ad hoc networks which means the network is expected to work and maintain without manual management or configuration.

2.1.2.3 MANET and its Related Networks

Since IETF defined MANETs in 1999, MANETs have many applications and implementations under the conceptual block of being an ad hoc network. However, some of the implementations have their own specific applications, deployments, challenges , and thus, have become another field of interest to many researchers. These extended families of MANETs, for examples, are Wireless Sensor Networks (WSNs), Wireless Mesh Networks (WMNs), Vehicular Ad hoc Networks (VANETs) etc. The following gives a brief discussion about these important relevant networks, in order to understand the common and the different characteristics between them.

Wireless Sensor Networks (WSNs) It was in the 1990s, when wireless technologies and low-power VLSI design became feasible, that researchers began envisioning and investigating large-scale embedded wireless sensor networks for physical environment sensing applications. The desire of researchers was to do computation and control environments. WSNs consist of individual nodes that are able to interact with their environment by sensing or controlling physical parameters. These nodes have to collaborate to fulfill their tasks as a single node is in capable of doing so. The collaboration between nodes happens via wireless communication (cheaper and more flexible than wired communication). There exist many important applications relying on WSN, such as, disaster relief applications, Intelligent buildings, Environment control and biodiversity mapping, etc. Because of its flexibility, there is not a single set of requirements for classifying all WSNs. However, those applications share some basic characters of WSNs. In most of them, there are plenty of normal sensor nodes and a few of sink nodes, where data from sensor nodes should be delivered to.

The very first point of difference between MANETs and WSNs is the application and equipment. In general, WSNs are deployed in a very large geographical areas, the equipment meant to be cheap enough to have thousands or perhaps hundreds of thousands of entities. Furthermore, entities of WSNs also have a specific set of sensing tasks. On the other hand, nodes in MANETs are meant to fit with a wide variety of applications, such as disaster relief operations, firefighters communication, multi-media, web-browsing, gaming, etc. Thus, nodes in MANETs are quite powerful (such as laptop, PDA, smart phone) in order to fit with those applications. The more powerful of terminal in MANETs' applications also refer to battery life. It is a serious concern regarding battery life in WSNs since the terminal has less intervention with human. Hence, the maintenance, recharging or replacement, of battery is very rare in WSNs. Mobility is another issue that differs between WSNs and MANETs. Since WSNs have to interact with the environment, their traffic characteristics can be expected to be very different from human-driven forms of networks. Long periods of very slow or no mobility can change suddenly by an unexpected natural event. Thus, in the sense of being ad hoc, sensor nodes also need to practice self-organization, the same as nodes in MANETs. Another aspect of mobility to be considered in WSNs is the sinks of information process in the network. In order to gather information, the sink node can be, but not often, a mobile node. Actually, the mobility of a group of mobile nodes in MANETs moving in a related manner, stick together, may correlate to those mobility of MSNs.

In summary, WSNs are application specific and usually have slow to no movement, no human intervention and no battery change/refill.

Vehicular Ad hoc Networks (VANETs) A Vehicular Ad hoc Network (VANET) is a form of wireless ad hoc network to provide communications among vehicles and nearby road-side equipments. The very first application was about safety and efficient transportation. For this purpose, many projects have emerged such as Intelligent Transportation Systems (ITS), Dedicated Road Infrastructures for Vehicle safety in Europe (DRIVE), Program for European Traffic with Highest Efficiency and Unprecedented Safety (PROMETHEUS) etc. In 2007, Car 2 Car Communication Consortium [18] has been initiated by European vehicle manufacturers, such as BMW, Audi, Opel, etc. Sharing point of those programs are the studying of Inter-Vehicle Communication (IVC) or Vehicle-to-Vehicle (V2V) communication which have formed and become, today, the popular VANETs. With the Internet era, the second main applications for cars arise, that is entertainment purpose applications, such as sharing multi-media and providing services based on Internet between cars. Actually, network architectures for VANETs can be Cellular networks, WLAN, pure Ad Hoc Network or Both (Hybrid networks). However, the existing applications found in literatures are mutually focusing more on hybrid networks. The summarizing of VANETs' main characteristics can be found as follows. The first characteristic of VANETs is the highly dynamic topology. Not only because of the speed of car but also the fact that cars are driving in opposite directions which makes topology changing even more quickly. For example, assume that the wireless transmission range of each vehicle is 100 meters. If two cars with the speed of 70km/hrs (19.5 m/s) are driving in opposite directions, the link will last only for at most 5 seconds. Accordingly, VANETs has frequently disconnected network. According to the safety application, the VANETs often have a new type of communication that addresses geographical areas where packets need to be forwarded. Thus, many researches on routing protocol proposed algorithms based on position or geo-location of nodes, such as Greedy Perimeter Stateless Routing (GPSR) [19], Geographic Source Routing (GSR) [20] and Anchor-based Street and Traffic Aware Routing (A-STAR) [21]. However, the utilization of geographical and position in VANETs also means that the protocol needs an existing street map or geology map pre-installed in the system or an aid from Global Positioning System (GPS). GPS provides accurate location and time information by utilizing satellites orbiting the Earth, monitoring stations on Earth and the GPS receivers owned by users (i.e., TomTom, Garmin nüvi, Magellan RoadMate etc.). By using GPS, a protocol cannot deny the utilization of existing technology and thus the communication is based on hybrid networks. In general, car can provide higher power to the communication nodes, and thus, power consumption issue has less important in VANETs. Moreover, with the higher power, nodes can have higher capacity of processing too. This is where the multimedia applications become an interest to VANETs.

In summary, VANETs work in a highly dynamic topology. The mobile nodes of VANETs are generally powerful (i.e., higher battery and processing capacity, multiple interfaces for varying transmission/receiving capabilities, and equipped with GPS).

Wireless Mesh Networks (WMNs) The idea of providing wireless Internet to large numbers of people became very appealing and being increasingly deployed as Wireless Mesh Networks (WMNs). WMNs provide users with extended coverage for wireless Internet access while minimizing the infrastructure cost. Actually, a WMN operates just like a network of fixed routers, except that they are connected only by wireless links. By incorporating with WLAN, a WMN is an inexpensive way to provide *last mile broadband Internet access* [22].

Mesh networking provides a number of benefits to an existing WLAN as follows. First, since the wireless equipments are relatively cheap in the past few years, hence, a WMN enables rapid deployment with lower-cost backhaul. Second, it is easy to provide coverage in hard-to-wire areas. Third, a WMN has properties of self-healing (maintenance of the network), resilient and extensible. For resilient issue, a mesh networking has an intention to use redundant and distributed nodes to provide greater reliability and range for any given WLAN. The overall scenario of WMNs, thus, comprises of some nodes connected to Internet via physical wires (of WLAN or Cellular Network), while the remaining nodes access the Internet through these wired gateways by forming a multi-hop WMN with them. Generally, wireless mesh networks (WMNs) are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad hoc network and maintaining the mesh connectivity [23]. In practice, the multi-hop WMN composes of many mesh routers and mesh clients. The conventional mesh clients only connect wirelessly through a mesh router with gateway/bridge. Mesh router contains the routing capability for gateway/bridge functions as in a conventional wireless router and additional routing functions to support mesh networking. These mesh routers have minimal mobility and form the mesh backbone for mesh clients. Such configuration is called conventional WMN [23].

On the other hand, the architecture can be a combination of conventional WMN (mesh router with conventional mesh clients) and client meshing. The client meshing provides peer-to-peer network among client devices. In this type of architecture, a client node is as comparable as a normal node in ad hoc networks, that is, it has to perform additional functions like relaying messages and self-configuring. Thus, WMNs diversify the capabilities of ad-hoc networks instead of being another type of ad hoc networks.

The big conceptual differences between WMNs and MANETs is twofold. Firstly, nodes in a WMN mostly are fixed and therefore, topology changes are infrequent and occur only due to occasional node failures, node shut-down for maintenance or addition of new nodes. Secondly, the traffic flow characteristics do not change very frequently. This permits the optimization of network based on traffic measurement and profiling. As a consequence, proactive discovery of paths is preferable to obtain an effective backbone in WMNs. In contrast to MANETs, reactive routing strategies are a normal procedure due to mobility. Even said that, reactive strategies also apply in WMNs for the case of destination is inside the mesh [24]. IEEE802.11s [25] is a draft amendment for mesh networking, defining how wireless devices can interconnect to create a WLAN mesh networks. Although, the standard of mesh networks is not finalized yet, the current draft 5.0 has been approved with 89% with 421 comments [26]. The aim of the task groups is to finalized the standard within year 2010.

Consequently, instead of being another type of ad hoc networking, WMNs diversify the capabilities of ad hoc networks.

2.1.3 Delay-Tolerant Mobile Ad Hoc Networks (DT-MANETs)

Before we describe the Delay-Tolerant Mobile Ad Hoc Networks that is used throughout this manuscript, a brief review of Delay Tolerant and Disruption Tolerant Networks is presented in the following sub-section for clarity.

2.1.3.1 Delay Tolerant and Disruption Tolerant Networks (DTNs)

Delay Tolerant and Disruption Tolerant Networks (DTNs) is more commonly known as Delay Tolerant Networks now as the problem of ‘delayed messages’ / ‘delayed packets’ is more prominent and subsumes the disrupted messages as well. DTNs address networking environments where continuous end-to-end connectivity cannot be assumed. In other words, DTNs are related to interconnection of highly heterogeneous networks together even if end-to-end connectivity may never be available. Environment examples include spacecraft, military area, some forms of disaster response, underwater, and some forms of ad-hoc sensor/actuator networks [27].

In April 2007, IETF RFC4838 [28], ‘Delay-Tolerant Networking Architecture’, has been published. Its description is briefly explained as follows. The architecture includes the concepts of occasionally-connected networks that may suffer from frequent partitions. The basis for this architecture lies with that of the Interplanetary Internet, which focused primarily on the issue of deep space communication in high-delay environments. However, according to the RFC, they expect the DTN architecture to be utilized in various operational environments as stated above. RFC4838 also defined an end-to-end message-oriented overlay called the ‘bundle layer’, situated at a layer above the transport and below applications. The bundle layer forms an overlay that employs persistent storage to help combat network interruption. In general, DTN architecture provides services similar to electronic mail but with enhanced naming, routing, and security capabilities. These enhanced features make DTN able to make progress towards the destination, even when no contemporaneous route exists. The RFC stated more about the ability of nodes that if nodes unable to support the full capabilities required by the architecture, nodes maybe supported by application-layer proxies acting as DTN applications.

The concept of DTN is employed widely in WSN, (i.e., [29], [30]) , and MANET fields (i.e., [31]). In the context of this work, we particularly interested in DTN characteristics found in MANET environment and thus, we present the definition, characteristics and challenges of DT-MANETs in the following subsection.

2.1.3.2 Definition, Characteristics and Challenges of DT-MANETs

Facing to the reality, ad hoc networks are often vulnerable to network partitioning. Node mobility, physical obstacles, limited radio range, or severe weather may prevent nodes from communicating to each others and keep the network in partitioned state. According to the definition 8, IETF RFC2501 does not mention the partitioning of the MANET in particular. Thus, we define Delay-tolerant mobile ad hoc networks (DT-MANETs) as follows:

> **Definition 10** (Delay-tolerant mobile ad hoc networks).
> *Delay-tolerant mobile ad hoc networks, shortly denoted as DT-MANET, constitute an emerging relative class of MANETs that feature frequent and long-duration partitioned MANETs. In other words, the main characteristic of a DT-MANET is the presence of multiple connected components composing a MANET.*

A DT-MANET is basically a combination of main key characteristics of DTNs and MANETs, which are the partitioning, mobile and ad hoc networks. The characteristic of DT-MANET as described above is similar to those networks called highly-partitioned wireless ad hoc network [32], sparse mobile ad hoc networks [33] and disconnected mobile ad hoc networks [34]. The latter proposes the definition of 'Opportunistic Networks' as follows. In opportunistic networks, mobile nodes are enabled to communicate with each other even if a route connecting them never exists (To this point, it is the same idea as DTNs and thus, DT-MANETs). Furthermore, nodes are not supposed to possess or acquire any knowledge about the network topology, which (instead) is necessary in traditional MANET routing protocols. Routes are built dynamically and are provided based on the idea that it is likely to bring the message closer to the final destination. This means while messages are in route between the sender and the destination(s), any possible node can opportunistically be used as the next hop.

Regardless of the name, in summary, DT-MANET is a challenged environment of MANETs where end-to-end connectivity cannot always be obtained. Such environment can be compared as a disconnected or partitioned MANETs. In this platform, usually the messages can be stored temporary on stations and are forwarded later when the situation allows to do so. Key problem of such network is the successful communication.

Examples of applications in such networks are vast, since in reality we can found many partitioned network such as many geographically groups of car on highway, shopping mall scenario, many teams of camper inside a reservation park, etc. Similarly to the original reason of why MANETs become of greatly interest, DT-MANETs can help the communication when is needed, such as lacking of infrastructure, reducing cost occurring in using existing infrastructure, etc. Generally, DT-MANETs addresses a more challenging problem than

MANETs and a solution to DT-MANET problem is usually applicable to the same/similar problem in MANETs too, but not vice-versa.

From the description above, Figure 2.4 depicts a concluded relationship between these families. In the scope of wireless ad hoc network, DT-MANET addresses the particular feature of partitioned network. Any protocol that can withstand this partitioned feature works also in a connected network of MANETs. Meanwhile DT-MANETs and MANETs are the communication conceptual which has the differences of partitioning or not of a network, other related networks (VANETs, WSNs and WMNs) are more specific in application and implementation details. VANETs mainly deal with mobile nodes, while WSNs and WMNs are likely to have more static nodes inside their network. However, there exists overlapping between these set of networks in real world applications as described previously. On the right hand side of Figure 2.4 depicts the overlapping of these networks. The mobility of nodes is likely to decline from VANETs to WMNs. Meanwhile, all these related networks can practically setting in both DT-MANETs and MANETs.

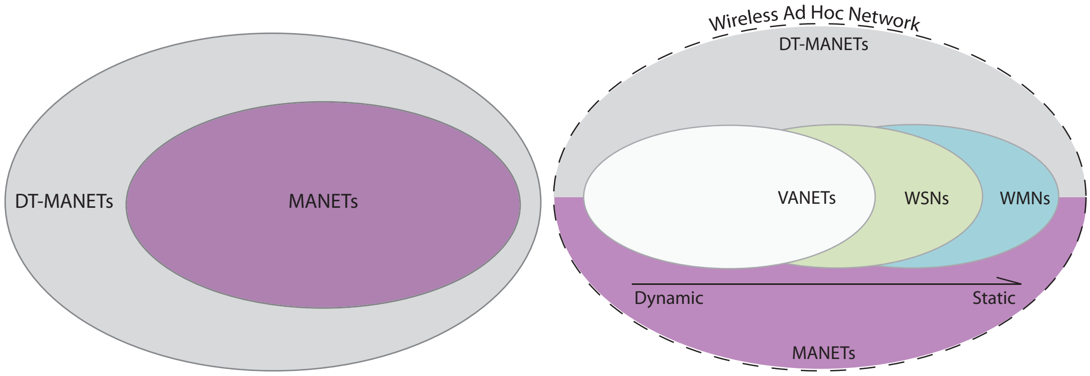


Figure 2.4: Relationship between DT-MANETs, MANETs, WSNs, WMNs, and VANETs

2.1.4 Modeling of an Ad hoc Network

Generally, there exists four different methods for evaluating the performance of a networking system: analytical, simulation, emulation, and testbed experiments. Analytical methods, however, are limited to very simple network configurations. Therefore, they are of limited use for complex networks and protocols. Using network simulators for performance evaluation is a trade-off speed for accuracy (the more detailed the models, especially at the lower layers, the slower the simulation). However, with their convenience for quickly exploring a large number of elements makes them, by far, the most common method of performance evaluation in the literature.

Testbed experiments are the most realistic method, but using the real devices and installing in the real environment are expensive and difficult to manage. Network emulation is typically somewhere between simulations and testbeds in terms of convenience and accuracy. Basically, emulator use the output of the wireless cards as inputs and emulate the wireless transmissions between nodes and control the movement of nodes in simulation based as if they are interacting inside specified virtual environments. In this study, simulation is chosen as a mean for evaluation since it saves a lot of cost and time in accomplishing the evaluation.

Definition 11 (Model).
Model is an abstract representation of something that only contains the details which are important for the consideration. A model typically is application-dependent. It may happen that the same thing is modeled different for different considerations [35].

Generally, different research works study by using different models, such as the differences in detail of elements, configurations and mobility models of ad hoc networks according to their interest. In this section, we will discuss overview of the main characteristics for modeling ad hoc networks.

2.1.4.1 Graph Modeling

In order to model ad hoc networks, graph theory is often used. Varieties of specific type of graphs representing networks can be found in the literatures such as euclidean distance graph, neighboring graph, unit-disk graph, connection graph and so on. In the context of our study, we are interested in communication graph which can be defined as follows:

Definition 12 (Communication Graph).
Relatively to a DT-MANET, a communication graph is an undirected geometric graph $G = (V, E)$. V is the set of vertices corresponding to the network stations, each vertex is characterized by a set of euclidian coordinates. E is the set of edges materializing stations' neighborhood. An edge $e = (u, v)$ exists if and only if the euclidian distance between vertices u and v is lower than a given threshold corresponding to the transmission range of each other.

According to the definition above the definition of one-hop neighbors in communication graph points of view can be given below.

Definition 13 (One-hop Neighbors – in Communication Graph).
For all edges (v_i, v_j), the stations associated to v_i and v_j are said to be one-hop neighbor to each other.

According to what we discussed before, many researches dealing with ad hoc networks assume static graphs. This is valid for many applications such as WSNs for fire detection inside building and WMNs in the part of mesh router. However, wireless ad hoc networks, MANETs and DT-MANETs are different. These kinds of networks address directly to the moving ability of stations. Actually, this mobility characteristic can represent in graph theory by adding time function into the communication graph. Thus, we define the dynamic graph as follow.

Definition 14 (Dynamic Communication Graphs).
Relatively to a DT-MANET, a dynamic communication graph is an undirected graph $G(t) = (V(t), E(t))$*. For each* t*,* $G(t)$ *is a communication graph.* $V(t)$ *is the set of vertices corresponding to the network stations at time* t*, and* $E(t)$ *is the set of edges materializing stations' neighborhood at time* t*. An edge* $e(t) = (u(t), v(t))$ *exists if and only if the stations corresponding to vertices* u *and* v *are in communication range of each other at time* t*.*

For the dynamic graph, the time dimension has been added into the communication graph and the set V and E may change over time. In other words, dynamic communication graph is a series of static communication graphs. At time t, the static communication graph represents a snapshot of the DT-MANET.

Up until now, there is no definition about connectivity in dynamic communication graph. Generally, in a connected component, there exists a path from each vertex to every other vertex. Definition 15 defines the connected component characteristic of a dynamic communication graph.

Definition 15 (Connected Component of a Dynamic Communication Graph).
Given $G(t)$ *a dynamic communication graph. The connected components belonging to* $G(t)$ *at time* t *are denoted by* $P_i(t)$*, such that, for any* $t, G(t) = \bigcup_{1..m} P_i(t)$*.*

So far, we present loosely definitions in graph theory regarding communication graphs that are used in modeling ad hoc networks. However, to be more specific, researches in ad hoc network field have high interest in the trajectory patterns of mobile stations, which include location, speed, direction, communication range between two nodes and some particular characteristics over simulation time. All these behaviors of mobile stations are represented by 'mobility models'.

2.1.4.2 Mobility Models

Mobility models are used heavily for studying and evaluating performance of algorithms for ad hoc networks under simulation method. To design and select a realistic mobility model that truly depicts and predicts nodes' mobility in ad hoc network draw a significant amount of research interest. Nowadays, this is another big research field.

Definition 16 (Mobility Models).
Mobility Model is a model that describes the movement of mobile nodes. Particular in simulating field, mobility model is a method of simulating movement of mobile nodes, usually for the purpose of further using the resulting movement for other simulations.

There exists a large number of mobility models used in literatures. Generally, these models can be classified based on their realism and complexity. So, we can classified them into two majority types: real traces and synthetic models. A real traces mobility model is a very accurate model since it captures the real mobility patterns from a particular scenario. Utilizing such a strict model can prove the ability of a protocol for a niche problem. However, it cannot verify the performance of a network protocol in general scenario case. On the other hand, the synthetic mobility models usually provide higher diversification of scenarios: including different types of mobile nodes (i.e., devices on vehicles, devices on people.) and environments (i.e., city center, university campus, highway, shopping mall). The synthetic mobility models can also be classified further depending on their realism and complexity. Sichitiu [36] gives a good review on the classification of mobility models. Many mobility models has been proposed in literatures. The most basic model is random waypoint (RWP) mobility model [37]. It is a stochastic one and have very least complexity since the operation is based on randomness. More realism models have been proposed in different scenario and complexity such as Freeway [38] (a model for vehicle movements), Column [39] mobility model (modeling the behavior of soldiers or a search troop), etc. Human mobility model (HMM) of [40] is a generic mobility model that represents the intention-driven mobility of people in metropolitan areas and also provides different scenario modeling such as Mall, City-center, and Highway mobility models.

In general, mobility models generate a set of resulting movement information over time dimension which can be used/replayed by other simulations. Normally, mobility models already take into account the environment information such as obstacles, trajectory maps and signal interference. Thus, the set of resulting movement information are the result after processing all those obstacles already. The final traced file may comprises of nodes, edges, distances between nodes, signal strength of edges and the changes between time t_i to t_{i+1}. Later, this traced file are used by other simulators for evaluating networking protocol.

To date, mobility model is a research area which still requires development of the better model to represent different scenario cases. Since the more complex and accurate the model is, the more simulation time is required. Hence, there is always a trade-off in selecting suitable mobility models and this issue is very much application dependent.

2.2 Multi-Objectives Optimization Problem (MOOP)

2.2.1 Introduction

Definition 17 (Optimize – Cambridge Advanced Learner's Dictionary).
to make something as good as possible

In mathematics and computer science, optimization, refers to choosing the best element from some set of available alternatives. More generally, optimization means finding 'best available' values of some objective functions given a defined domain, including a variety of different types of objective functions and different types of domains. The terminology and definitions are described in more details in [41].

Definition 18 (Instances of an optimization problem).
An instance of an optimization problem is a pair (F, c), where F is any set, the domain of feasible points; c is the cost function, a mapping

$$c : F \rightarrow R^1$$

The problem is to find an $f \in F$ for which

$$c(f) \leq c(y) \text{ for all } y \in F$$

Such a point f is called a globally optimal solution to the given instance, or, when no confusion can arise, simply an optimal solution.

Definition 19 (Optimization problem).
An optimization problem is a set I of instances of an optimization problem.

According to the latter definition, the optimization problem can have multiple instances which means there are multiple cost functions and the corresponding domain of feasible points.

When an optimization models a system involves only one objective function, the task of finding the optimal solution is called single-objective optimization. Well-known examples of single-objective problems include minimum spanning tree, traveling salesman problem, minimizing the value of a linear function (e.g., $c'x = c_1x_1 + c_2x_2$ of a particular domain, subjects to a set of constrained functions), etc.

A multi-objective optimization problem (MOOP) differs from a single-objective optimization problem as it involves several objectives that require optimization. According to the Definition 18 and 19, MOOP add more instances $c(f) \leq c(y)$ for all $y \in F$ to set I of the optimization problem as the objectives increase. When optimizing a single objective problems, the best single design solution is the goal. However, for multi-objective problems, with several (possibly conflicting) objectives, there is usually no single optimal solution. MOOP presents a possibly infinite set of solutions, which when evaluated, produce vectors whose components represent trade-offs in objective space. Therefore, a compromise is required in decision making to select a solution (or solutions) from a finite set by selecting one or more of these vectors. A good discussion about MOOP can be found in [42], [43], [44], etc. For this study, MOOP is mathematically defined in the next section.

2.2.2 Related Definitions to MOOP

Definition 20 (Multi-Objective Optimization Problem).
In general, a MOOP minimizes or maximizes $F(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x}))$ *subject to some constrained functions* $g_i(\vec{x})$, $i = 1, \ldots, m$, $\vec{x} \in \Omega$. *An MOOP solution minimizes/maximizes the components of a vector* $F(\vec{x})$ *where* $\vec{x}$ *is an n-dimensional decision variable vector* ($\vec{x} = x_1, \ldots, x_n$) *from some universe* Ω

In general, each objective function f_k of a MOOP can be taken as minimizing or maximizing problems, and thus, there exists a combination of minimizing one objective while maximizing another objective in a MOOP. For simplicity, we shall only give definitions with the case of minimization problem in this chapter from this point onward. There are some basic MOOP definitions which will be used later on within this dissertation and thus, we shall present them now.

Definition 21 (Pareto Dominance).
A vector $\vec{u} = (u_1, \ldots, u_k)$ *is said to dominate* $\vec{v} = (v_1, \ldots, v_k)$ *(denoted by* $\vec{u} \preceq \vec{v}$*) if and only if* u *is partially less than* v*, i.e.,* $\forall i \in \{1, \ldots, k\}, u_i \leq v_i \wedge \exists\, i \in \{1, \ldots, k\} : u_i < v_i$. *[44]*

Definition 22 (Pareto Optimality).
A solution $x \in \Omega$ *is said to be Pareto optimal with respect to* Ω *if and only if there is no* $x' \in \Omega$ *for which* $\vec{v} = F(x') = (f_1(x'), \ldots, f_k(x'))$ *dominates* $\vec{u} = F(x) = (f_1(x), \ldots, f_k(x))$. *The phrase 'Pareto optimal' is taken to mean with respect to the entire decision variable space unless otherwise specified. [44]*

Definition 23 (Pareto Optimal Set).
For a given MOOP $F(x)$*, the Pareto optimal set* ($\mathcal{P}^*$) *is defined as:*

$$\mathcal{P}^* := \{x \in \Omega \mid \neg\exists\; x' \in \Omega\;\; F(x') \preceq F(x)\}. \;[44]$$

Definition 24 (Pareto Front).
For a given MOOP $F(x)$ *and Pareto optimal set* $\mathcal{P}^*$*, the Pareto front* ($\mathcal{PF}^*$) *is defined as:*

$$\mathcal{PF}^* := \{\vec{u} = F(x) = (f_1(x), \ldots, f_k(x)) \mid x \in \mathcal{P}^*\}. \;[44]$$

The Pareto optimal solutions are ones within the search space whose corresponding objective vector components cannot be improved simultaneously. Their corresponding vectors are known as *nondominated*. Selecting a vector from this Pareto Front set $\mathcal{PF}^*$ implicitly indicates acceptable Pareto optimal solutions. The nondominated vectors, when plotted in criterion space, is known as the Pareto front.

2.2.3 Example Case and Multi-Objective Optimization Procedure

Let us consider the decision-making involved in buying a mobile phone. Accordingly, there are so many choices for customers, ranging from very cheap and simple one (choice 1) to a very expensive one but very powerful also (choice 2), as shown in figure 2.5. If the cost is the only objective of this decision-making process, then the optimal choice is the Choice 1. Unfortunately, customers are not as simple as that, otherwise there will be no so many phones with different technology and features in the market. In general, it is expected that an expensive phone comes with a higher technologies or more features than a cheap one. The figure illustrates that the technology&features level of Choice 1 is 5 percent comparing to the phone of Choice 2. For people who are rich enough and demand the latest or the best technologies at the time will definitely select the Choice 2. On the other hand, for people who do not care about features, but require just a phone that can do voice communication, then the Choice 1 is an optimal solution. Between these two extreme solutions, there exist many other solutions, which are a trade-off between cost and technologies & features. A number of such solutions (solutions A, B, and C) with differing costs and technology & features levels are shown in the figure. Thus, between any two such solutions, a particular solution is better in terms of one objective, but inevitably compromises other objectives.

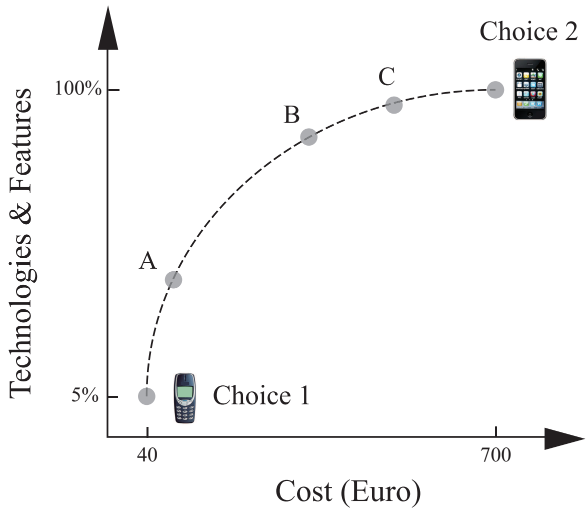


Figure 2.5: Hypothetical trade-off solutions for acquiring a mobile phone

The example above is indeed a two-objective optimization problem (sometimes is called bi-objective optimization). For two conflicting objectives, each objective corresponds to a different optimal solution. In this case, Choice 1 and 2 are these optimal solutions. Now let consider the solution A, B, and C. A customer cannot choose the best solution among these three choices with respect to both objectives, because there are always a trade-off between them. In other words, without any further information, there is no best solution from this set that make both objectives look better than any other solutions. All solutions shown in Figure 2.5 are optimal solutions for both objectives.

Indeed, the multi-objective optimization gives us all the optimal solutions, but in reality the customer still want to buy only one mobile phone. In order to make final decision,

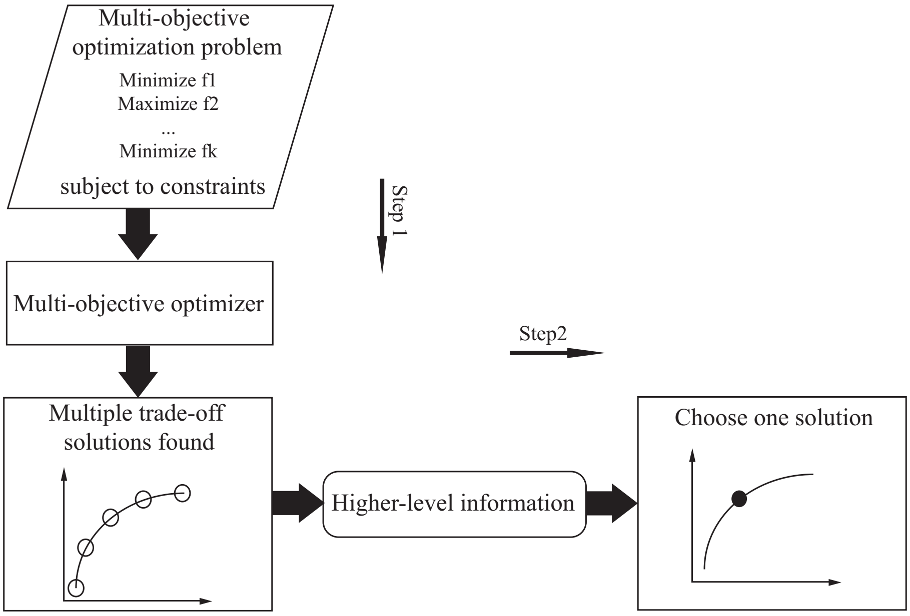


Figure 2.6: Multi-objective optimization procedure

the customer normally considers other factors such as per month contract payment, the durability of the mobile phone, the design, shape, color and many other factors. These, so called, higher-level information are often non-technical, qualitative and experience-driven [43]. All these processes happen in finalizing the decision and are called multi-objective optimization procedure. Figure 2.6 [43] depicts this procedure. Step 1 is concerned the most in the multi-objective optimization field. Actually, it presents the core process of multi-objective optimization and the goal is to find as many Pareto optimal solutions as possible in a problem.

2.2.4 Optimization Algorithms

Finding optimal solution(s) can be done using many approaches depending on types of problem. While the differences between single objective and multi-objective optimization problems has been introduced under Section 2.2, the primitive single objective problem also can be differentiate further with level of computational complexity (from polynomial to NP-complete problems). Further details of computational complexity can be found in [45]. However, we give a short description of our discussion here for better understanding later as follow. For a single objective problem that is P problem means it is solvable in polynomial time by some algorithms, for example, the optimal solution of minimum spanning tree problem can be found by greedy algorithm like Kruskal or Prim's algorithm, [46] and [47] respectively. However, combination of multiple P problems (with conflicting properties between objectives and becoming multi-objectives problem) is a totally different field of solving problem and, in general, the difficulty of a multi-objectives problem cannot be concluded based on each objective's difficulty. In this case, combination of multiple P objective problems may or may not be solvable in polynomial time using exact algorithm. In other words, the multi-objective problem is a non-polynomial problem if it contains an NP-hard problem among other problems. The difficulty level of problem usually suggests the types of optimizer (i.e., optimization algorithm).

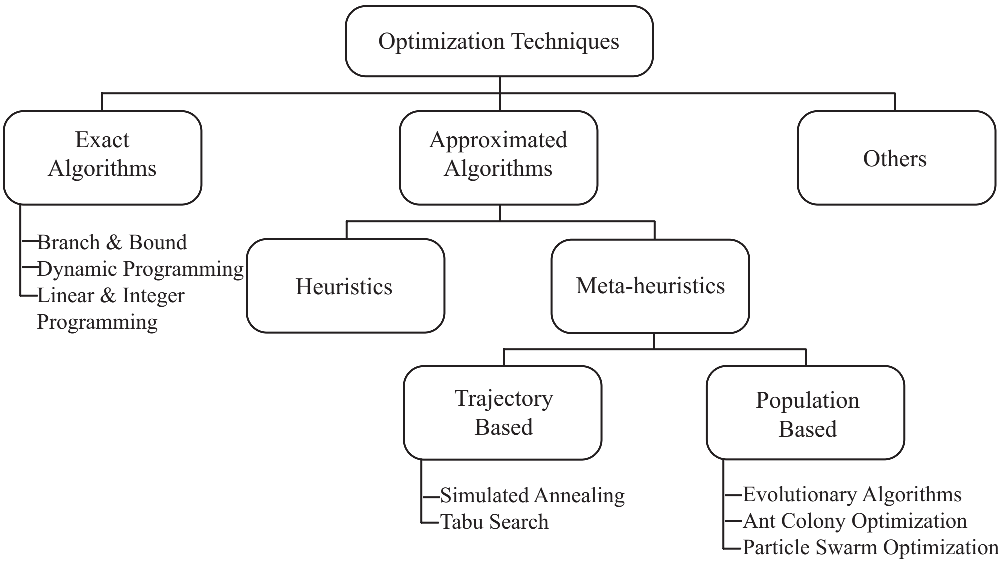


Figure 2.7: Taxonomy of Optimization Techniques [1]

Figure 2.7 gives us a basic taxonomy of optimization techniques. In literature, optimization techniques or algorithms can be classified roughly into 'exact approach' an 'approximated approach'. The exact algorithms can guarantee to find the optimal solution for all the finite instances, but they require a large amount of run-time to solve complex problems like NP-hard or problems with very big search space (e.g. exponential search space). Exact algorithms, thus, can be practically applied only small or moderately sized instances. Hence, big search space and high complexity problems require approximated algorithms to find the best optimality set instead. However, the guarantee of finding true optimality solutions has to be compromised for good enough solutions due to time constraint. Examples of exact algorithms are, such as, branch-and-bound, dynamic programming, linear and integer programming [48]. The approximated approach consists basically of basic heuristic methods (approximated techniques with stochastic guided components) and meta-heuristics method. Blum [49] provides some meta-heuristic definitions, but in general we can state that meta-heuristics are high level strategies having a given structure that specifies the application of a set of operations (variation operators) to explore high dimensioned and complex search spaces. In fact, the classification of meta-heuristics can be done in many different ways [49]. Following Figure 2.7, under the meta-heuristic approach, taxonomy are classified into 'Trajectory Based' and 'Population Based'. In general, the algorithms of trajectory based approach are simpler, they find only one best solution per computing round, but the main drawback is the danger of converging to a local optima. Examples algorithm in this trajectory based are simulated annealing [50] and tabu search [51], etc. On the other hand, the population based algorithms solve problems by simultaneously producing a set of solutions, so called population, in each iteration. The use of population solutions makes it easier to diversify the search so that solutions can or may approach the global optima. If an optimization problem has a single optimum, all population solutions can be expected to converge to that optimum solution. However, if an optimization problem has multiple optimal solutions, the population based algorithms can provide multiple optimal solutions in its final population. Examples of algorithms in this class are evolutionary algorithms [43], ant colony optimization (ACO) [52], particle swarm optimization (PSO) [53] and scatter search [54], etc.

2.3 Summary

In this Chapter, the overview of ad hoc network and its variations are discussed. In particular, we give the definitions and examples of different types of ad hoc networks. This study focuses on DT-MANET due to its relatively close to reality with characteristics such as network has changes over time, sometimes being sparse or dense, and can have different partitions. DT-MANETs can be described using a dynamic and partitioned communication graph. Movement of stations can be modeled by means of different mobility models.

The presentation of DT-MANET in this study utilizes those components of Graph Theory. Hence, the related definitions are introduced. As one of the objectives of this study involves finding a solution to a what can be seen as multi-objective optimization problem (MOOP), the basic concepts of MOOP are introduced. A simple example of MOOP is illustrated and their possible solutions are discussed.

Chapter 3

Topology management in Ad Hoc Networks

Contents

3.1 Introduction **30**

3.2 Relevant Topologies in MANETs **30**

3.3 Topology Management **32**

3.3.1 Motivation 32

3.3.2 Different Aspects of Topology Management 33

3.3.2.1 Physical Topology Management 33

3.3.2.2 Logical Topology Management (Virtual Backbone) 34

3.3.3 Mobility Issues and their Influences on Topology Management Algorithms 37

3.3.3.1 Centralized Algorithms 38

3.3.3.2 Distributed and Decentralized System 38

3.3.3.3 Desirable characteristics of algorithm for managing topology in DT-MANETs 39

3.4 Spanning Tree **40**

3.4.1 Motivation 40

3.4.2 Constructing and Maintaining Spanning Forest in MANETs 40

3.4.3 Dynamicity Aware - Graph Relabeling System (DA-GRS) 41

3.4.4 Problems Relating to Spanning Tree 43

3.4.4.1 Minimum spanning tree (MST) problem 43

3.4.4.2 The multi-criteria MST problem 44

3.5 Robust and Efficient Topology Management in DT-MANETs . . **46**

3.6 Summary **48**

3.1 Introduction

Topology is a major area in mathematics. In mathematic context, there are so many branches of theory related to topology, such as topological graph theory and topological combinatorics. The definitions of topology are different from one theory to another and solely depending on the specific context. Although, topology are defined differently from one to another work, basically topology emerged through the development of concepts from geometry and set theory. Under the context of this work, ad hoc networks, we shall define our topology definition and discuss topology in the sense of networking patterns only.

Definition 25 (Topology – relevant to networking).
In the sense of computer network, topology defines the topological space, comprising of a basic set of elements of topology, which basically is the set V and E of communication graph G. Furthermore, the topology also defines geometrically the dimension, shape, relative position of such elements in space of G.

In general, topology has been used to describe physical topology. Our definitions of communication graph and dynamic communication graph found at Definition 12 and 14, respectively, are also described the availability of network components (vertices set, V and edges set, E) at physical topology level.

However, it is in nature of networking field that we discuss and abstract some facts from physical level to virtual or logical level. In a computer network, the topology describes either the physical or logical arrangement of the computing and networking elements.

Definition 26 (Logical (virtual) topology).
A logical or virtual topology of a communication graph $G = (V, E)$ is a subset $G' = (V', E')$ of graph G, denoted $G' \subseteq G$ where $V' \subseteq V$ and $E' \subseteq E$.

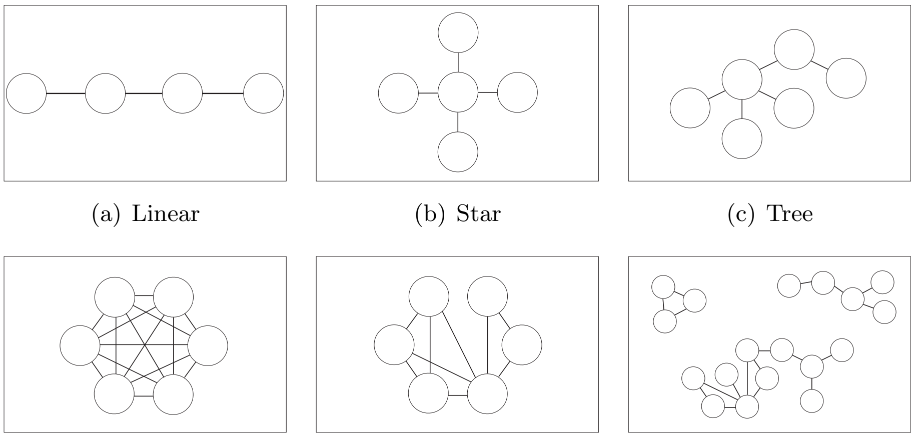
In a communications network, topology is the pattern of interconnection between nodes; for example, a linear list, star or tree configuration. We shall discuss briefly about these common topology patterns, especially for those found in MANETs in the following section.

3.2 Relevant Topologies in MANETs

Topology patterns have been captured and studied in field of communication networks because each pattern affects the networking protocol both in terms of difficulty and efficiency. There are several different topologies relevant to the wireless and mobile ad hoc networks. Thus, this work focuses only at the typical topologies found in MANETs and exclude some existing network topology such as hypercubes, grids, etc. The basic topologies in networking can be seen in Figure 3.1 and we describe their characteristics here shortly. Please note that each sub-figures in Figure 3.1 is a connected graph except the Sub-Figure 3.1(f) that is a disconnected graph or so called a partitioned graph.

Definition 27 (Connected Graph).
A graph is connected if there is a walk between any two vertices. A walk in a graph is an alternating sequence of vertices and edges, beginning and ending with a vertex, in which each vertex (except the last) is incident with the edge which follows it and the last vertex is incident with the preceding edge. It is closed if and only if the first vertex is the same as the last and otherwise open.

Definition 28 (Subgraph).
A subgraph of a graph G is a graph whose vertices and edges are subsets of the vertices and edges, respectively, of G.



(a) Linear (b) Star (c) Tree

(d) Full-connected (e) Partially Connected (f) Partitioned

Figure 3.1: Examples of Topology in MANETs

Linear The most primitive connecting type occurring between two wireless nodes is to form a line, linear list or string form. In such topology, all nodes have at least one connection with its neighbor and the set of 1-hop neighbors has only two neighbors at maximum. Multi-hop communication in this topology uses the collaboration among neighbors for relaying (forwarding) messages. It can be said that this topology is very common in MANETs especially in the case of highway mobility model (i.e., a caravan of cars traveling together on highway).

Star In a star topology, each network node (leaf node) is connected to a central node (hub node). With point-to-point connection between the central node and the other nodes, all traffic from any node except the central node is a two-hop communication. On the other hand, the central node can send broadcasting message to all nodes in the network. Sometimes, we call such star topology network as a single-hop star.

Tree A spanning tree T_G of a connected graph G can be defined as a maximal set of edges of G that contains no cycle, or as a minimal set of edges that connect all vertices of G. If the connected graph G comprises of n nodes, the number of edges equal to $n-1$. This tree has individual leave nodes which are required to transmit to and receive from one other node only and are not required to act as repeaters. Tree topology often is used in computer science ranging from a data structure to a hierarchy network topology type.

Full-connected, Clique or Complete graph A clique in an undirected graph $G = (V, E)$ is a subset of the vertex set $C \subseteq V$, such that for every two vertices in C, there exists an edge connecting the two. In fact, the subgraph induced by C is a complete graph.

Definition 29 (Complete Graph).
The complete graph on n vertices is that graph which has n vertices, each pair of which are adjacent.

This means example shown in Sub-Figure 3.1(d) depicts both a fully connected graph, each node within the network has a direct edge to every other nodes, and clique.

Partially connected topology Partially Connected topology is a connected graph but not a complete graph. This means that it exists a pair of vertices that is not adjacent.

Partitioned topology A topology graph for a mobile ad hoc network can have any arbitrary structure. Especially with DT-MANETs as presented in the previous chapter, the communication graph G can be partitioned into a collection of disjoint subgraphs, as shown in sub-figure 3.1(f).

3.3 Topology Management

Definition 30 (Topology Management (TM)).
Topology Management (TM) is of managing, handling, supervising or controlling the network topology in order to provide better control over network resources and to increase the efficiency of communication.

3.3.1 Motivation

In a relatively dense network, typical wireless networking problems are aggravated by the large number of neighbors such as the interfering between nodes, too large transmission power for close nodes and re-computation of routing protocols when small movements of nodes happened. On the other hand, in a sparse network, the interference decreases since each node shares medium with fewer neighbors. Instead, the difficulties come from network partitioning and the way of performing self-healing (maintenance process) after physical topology changed. Some of these problems can be overcome by topology management techniques. Instead of using whole set of the possible connectivity of a network, a deliberate choice is made to restrict a topology of the network. Then, the topology of a network is determined by the subset of active nodes and the subset of active links. Actually, such process is the same as creation of a subgraph in order to meet some particular requirements. This can be done at physical level or logical level. A more formal definition of topology management techniques in terms of graph theory is defined in the following:

Definition 31 (Topology Management Algorithm).
A topology management algorithm takes a graph $G = (V, E)$ *representing the network where* V *is the set of all nodes in the network and there is an edge* $(v_1, v_2) \in E \subseteq V^2$ *if and only if nodes* v_1 *and* v_2 *can directly communicate with each other and builds subgraph from* G *to a managed topology graph* $T = (V_T, E_T)$ *such that* $V_T \subseteq V$ *and* $E_T \subseteq E$.

3.3.2 Different Aspects of Topology Management

Several taxonomies of topology control are given in literature such as [55] and [56]. However, these taxonomy schemes do not cover the whole literature about topology control and management since their scheme applied to specific assumptions, such as the aspect of underlying network and the use-case application. Regarding only the varieties of factors used by TM protocol is already hard to comprehend the role and purpose of each protocol. Moreover, these protocols make different assumptions about the underlying network topology, environment, resources and capacities. This circumstance makes it extremely difficult to understand and compare different protocols. Hence, summarizing the taxonomy of the area is extremely hard. Instead, we summarize TM protocols into two different approaches.

The first approach, the network can be managed or changed at physical level. The result of such approach is a new configuration of network at the physical level. The second approach of TM protocol enhance the network by managing logical topology and does not changing any physical parameters of the network. We present the general ideas of different techniques for TM protocol, separating in these two approaches as follows.

3.3.2.1 Physical Topology Management

In literature, various TM protocols or techniques do fine-tuning physically to obtain the desire network topology. Techniques of such approach mostly deals with controlling power operations and also are called 'Topology Control'. The controlling power techniques for physically managing topology including:

- **Controlling the set of active nodes** ($V_T \subseteq V$): It is the cutting off power strategy which also uses in green IT aspect. The idea is by periodically switching off nodes with low energy reserves and activating other nodes instead, exploiting redundant deployment in doing so. Tian et al. [57] proposed the idea where node-scheduling scheme guarantees that the original sensing coverage is maintained after turning off redundant nodes. A similar idea is found in [58] proposed a schema for saving network resources by presenting the idea of backing up node. In the same work, the method for determining which sensor nodes should be turned off is also provided.

- **Controlling the set of active links** ($E_T \subseteq E$): This scheme controls the set of active links by changing transmission range, also called Dynamic Voltage Scaling (DVS). Instead of using all links in the network, some links can be disregarded and communication is restricted to crucial links. An example, found in WSN when nodes are static, is operated by controlling the reach of a node's transmissions – typically by power control, but also by using adaptive modulations or changing angle of transmitter. To be more specific, this case controls the transmitting ranges, power, and/or degree of radio transmitter in order to generate a network with the desired properties. There are many techniques proposed such as topology sparsing techniques using proximity graphs [59], common power protocol - COMPOW [60], and K-NEIGH protocol [61], etc. If the transmitting range is reduced then the emission power is decreased and the power consumption is reduced accordingly. Reducing transmitting range also reduce the number of connected nodes and thus may help reducing problem of high collision and interference problems.

3.3.2.2 Logical Topology Management (Virtual Backbone)

Instead of direct fine-tuning the physical topology, several TM protocols aim at constructing a managed topology subgraph T, which is also called 'Virtual Backbone', such that some desired properties are met. In this case, TM protocol tries to cope with the existing network topology and provide better desired properties at logical or application level. In general, the main properties of managed topology subgraph T or virtual backbone at logical level comprises of being connected, providing energy efficient and having low stretch factor. A topology management algorithm should not disconnect a connected graph G. In other words, if there exists a path in G between two nodes u and v, there should also be some path available for these two nodes in T (it does not have to be the same path). Providing energy efficient issue is a big factor where the optimization can be done at both physical and application levels. Typically, the solution T should provide the lowest energy consumption path for all communications occurs in graph G, or provides energy balancing method in order to extend the network lifetime. Another concerning feature is the shortest path connecting any two nodes in the subgraph that should not be much longer than the shortest path connecting them in the original graph. This aspect of path quality is captured by the stretch factor of the subgraph. Let T be a managed topology of G and let x and y be two vertices in G. The stretch of x and y in T, denoted $Str_T(x, y)$, is the ratio of the distance between x and y in T to their distance in G. Given that the distance between any x and y is the hop-count. Formally,

$$Str_{T,G}(x, y) = max_{x,y \in G} \frac{|(x, y)_T|}{|(x, y)_G|}$$

However, this particular desire has been proposed mostly to apply with static wireless network since it is not suitable in a very dynamic one, such as [62] generating compact routing with minimum stretch in fixed network. A bounded stretch factor for routing is provided lately by [63] for ad hoc wireless network, but nodes are static.

Particularly within this subsection, we present common logical TM techniques exist in literature. These techniques mainly selects a subgraph T, which is also called 'Virtual Backbone', of the entire network topology. In literature, definitions of virtual backbone is generally avoided by the authors. We found that the given descriptions of virtual backbone is vary and subjective from one work to another. For examples, Lin et al. [64] described a virtual backbone to exploit a hierarchical structure, while Liang et al. [65] suggested that flat network structure should be employed to allows balanced loading on the nodes an the links.

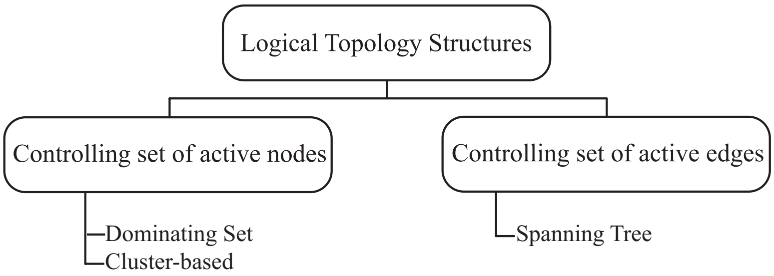


Figure 3.2: Taxonomy of logical topology structure or virtual backbone

Thus, we prefer to call such subgraph T as logical topology more than virtual backbone in this study. However, please note that the meaning of both terms are interchangeably used throughout this manuscript. Figure 3.2 depicts the taxonomy of logical topology or virtual backbone structure found in literature. General ideas of these structures and characteristics of techniques are presented shortly as follows:

Controlling set of active nodes Rearranging active links/neighbors into a hierarchical network topology where some nodes assume special roles. There are two examples in this category.

1. **Dominating Set and Connected Dominating Set**: First example is to select some nodes as a 'virtual backbone' for the network and to only use the links within this backbone and direct links from other nodes to the backbone. In this category, the backbone has to form, so called, a dominating set.

   Definition 32 (Dominating Set (DS)).
   Dominating Set (DS) is a subset $D \subset V$ such that all nodes in V are either in D itself or are one-hop neighbors of some node $d \in D$ ($\forall\ v \in V : v \in D\ \vee\ \exists\ d \in D : (v, d) \in E$).

   In other words, only the links between nodes of the dominating set or between other nodes and a member of the dominating set are maintained. Example works on DS for general graphs are the work of Haynes et al. [66] providing an extension treatment of domination issues, Fomin et al. [67] providing exact, exponential, algorithm for DS problem. Some works utilize DS to provide a better service in MANET, for an example, Stojmenovic et al. [68] providing broadcasting algorithms based on DS. Another related concept is Connected Dominating Set (CDS) also has been studied extensively as a TM structure. The formal definition of CDS is as follow:

Definition 33 (Connected Dominating Set (CDS)).
Connected Dominating Set (CDS) is a subset $D \subset V$ such that all nodes in V are either in D itself or are one-hop neighbors of some node $d \in D$ ($\forall\ v \in V : v \in D \vee \exists\ d \in D : (v, d) \in E$) and any node in D can reach any other node in D by a path that stays entirely within D. This means D induces a connected subgraph.

Basically, a CDS is a connected DS. Algorithms to create connected dominating set based virtual backbones are numerous and ranging from being centralized to distributed and asynchronous algorithms. One example on a centralized heuristic is [69] proposing a pruning-based heuristic. However, it was [70] that proposed a distributed algorithm based on two self-pruning rules. Later, Dai and Wu [71] proposed a generalization of the two previous rules, called as rule k, which reduces the time complexities from the first version. Another example works on CDS for MANETs is [72], providing a distributed and asynchronous way to construct and maintain CDS in a dynamic environment of MANETs using two-hop informations.

2. **Cluster-based**: This idea scheme is related to the first idea of DS and CDS with slightly different. In cluster-based topology management the network is partitioned into clusters. Clusters are subsets of nodes that together include all nodes of the original graph such that, for each cluster, certain conditions hold.

Definition 34 (Clustering).
Clustering consists in the assignment of a set of observations into subsets, called clusters, so that observations in the same cluster are similar in some sense.

The most typical problem formulation is to find clusters (clustering) with clusterheads, a representative of a cluster. In a clustered network, only links within a cluster are maintained (typically only those involving the clusterhead) as also selected links between clusters to ensure connectivity of the whole network. In summary, the clusterheads are responsible for managing communication between nodes within their own cluster as well as routing information to clusterheads of other clusters. In some works, the desired network will have each node situating only one hop away from its clusterhead. Problems in this topology management class fall for finding a suitable clusterhead, in terms of suitable number per network, quality of clusterhead node, and the geometrical position of clusterhead inside network. When the number of nodes in a cluster should be minimized, this is equivalent to finding a maximum (dominating) independent set which is an NP-Hard problem.

Definition 35 (Independent set).
Independent set is a subset $C \subset V$ such that $\forall v \in V$ $C : \exists\ c \in C : (v, c) \in E$ and no two nodes in C are joined by an edge in $E : \forall\ c_1, c_2 \in C : (c_1, c_2) \notin E)$.

Example works on cluster-based technique in MANETs are discussed as follows. In [73], a centralized clusterhead election algorithm is presented, where the base station assigns the clusterhead roles based on the energy level and the geographical position of the nodes. Some works such as [73], [74] used an unique identifier (ID) per node to compute its priority. All nodes locally exchanged these priority information within their two-hops neighborhood and then designated the clusterhead as the highest priority node. A well-known clustering algorithm called Weighted Clustering Algorithm (WCA) is proposed in [75]. WCA creates one-hop clusters with one clusterhead. A heuristic weight function combining the distances between the neighbors, the number of neighbors, the speed of the neighboring nodes and the remaining battery power of a node is proposed. Nodes are assumed to be provided with geographical information or relative distances of one node and its surroundings. Brust et al. proposed WACA in [76], an algorithm whose weight function relies solely on locally available information. The main concern of the authors in this work is to reduce the network communication overhead during the clustering process. As each node elects one neighbor as clusterhead in its direct neighborhood, their solution may lead to chains of clusterhead, i.e. a node x may elect y as clusterhead and y may elect z (x and z are not neighbors).

Controlling the set of active links/the set of neighbors for a node Instead of using all links in the network, some links can be disregarded and communication is restricted to essential links. An example of this idea is spanning tree Γ, where subset of $E_\Gamma \subseteq E_G$, is used as essential link with certain conditions hold. Given G a communication graph. Another example is the spanning forest, where several induced spanning trees are generated. Thus, V_G is partitioned logically into several sets of vertices and thus, each logical subgraph (i.e. spanning tree) has a smaller network to control. Examples under this scheme are discussed more in details in section 3.4.

3.3.3 Mobility Issues and their Influences on Topology Management Algorithms

Node mobility is a prominent feature of ad hoc networks [55]. In general, the mobility model is a key issue in managing topology. For maintaining the desired topology, the mobility increases message overhead. This has a negative effect on both network capacity and nodes' energy consumption. On the other hand, some studies claimed that mobility also has positive impacts on the network, such as improving routing capabilities [77]. However, this claim is true only if the network is connected where an end-to-end path exists between any two nodes in the network. Thus, this claim is an hypothesis that may not be true (hold) for DT-MANETs. There are also some other positive points of mobility described in the work such as mobility helps security, mobility assists information dissemination and mobility reduces uncertainty while building trust. However, the claimed benefits do not come directly from the mobility itself but rather from a specific algorithm that exploits some facts in mobility and then force the mobile node to do some actions, (such as move in close range to another node for building trust etc). Hence, mobility issue is still a major obstacle and need further study. The crucial issue is the relationship between mobility and applications when works with DT-MANETs.

Regardless the pros and cons of mobility, the issue defines the limitation and influence the characteristics of TM algorithms. This is discussed in the following section.

3.3.3.1 Centralized Algorithms

In centralized algorithm, the computation is done at a central entity which generally produce a solution S from the input I. For ad hoc networks, applying a centralized algorithm for topology management means (1) gathering the complete network topology (utilizing global information) into one network node so that it could (2) compute a solution for the whole network and then (3) disseminate its result back to all entities. Such a scheme is not suitable for dynamic network like MANETs or DT-MANETs mainly because of its mobility characteristic. We can generalize those unsuitable reasons as follows. First, this specific node has to be chosen or elected somehow. In a dynamic network, gathering the complete topology graph is impossible if the network has a large number of entities. The second problem is also about computing power and memory scalability. Ad hoc network nodes are generally characterized by limited computation capacities and memory. Computing a central algorithm on a large topology graph means the number of information is large and may not be able to solve within a single node. The third, disseminating the result back to all nodes needs a speed level that is able to compete with the topology changing rate. If the computing node managed to gather a valid topology at time t, it is also unlikely that the solution would be disseminated on time (i.e. at the time the solution reaches the nodes, the topology may have changed already). Therefore, dissemination of information is equally impossible as gathering the complete topology.

In summary, if the considered ad hoc network is a static and small network or having a very low mobility, then centralized algorithms can be used. Otherwise, a decentralized computation using distributed algorithms with asynchronous communication is prefered.

3.3.3.2 Distributed and Decentralized System

In a distributed system, the logical distribution of the functional capabilities is usually based on the following set of criteria:

- Multiple processes.
- Interprocess communication.
- Disjoint address spaces.

Distributed algorithm is an algorithm designed to run on computer hardware constructed from interconnected processors. In the other meaning, distributed algorithms are typically executed concurrently, with separate parts of the algorithm being run simultaneously on independent processors and having limited information about what the other parts of algorithm are doing.

Decentralized computing concept and distributed system are closely related. Basically, decentralized computing is the allocation of resources, both hardware and software, to each individual entity. There is no central management or a dedicated node to make decision. The decentralized computing also implies that the used information should not be the global information (having global information is somehow like the system own a central memory shared by all computing entities). Instead information should be provided locally such as k-hop information. As discussed in the previous section that using global information is almost impossible in dynamic network due to the failure in gathering and disseminating information process. Even if the gathering and disseminating information are done, the information can easily be out-of-date when needed. Thus, the utilization of local information is encouraged in purely decentralized system.

Asynchronous system has no global clock and it operates under distributed control with concurrent hardware components communicating and synchronizing on channels. In the context of this study, asynchronous communication plays an important role in order to exchange information and synchronize operations among collaborating nodes. A simple operation on channel consists of two tasks: a request and an acknowledge. Handshaking protocol is a common communicating operation in asynchronous systems. In a complete asynchronous communication, actions by the individual nodes may take an arbitrary but finite amount of time to complete. No specific assumption is made about whether the final decision is reached or successful.

One of the major challenges in developing and implementing distributed and decentralized algorithms is successfully coordinating the behavior of the independent entities as some of them may fail and the communication between them may be unreliable. Hence, designing a successful and efficient asynchronous communication is a very crucial task in topology management too.

For ad hoc networks, devices such as multiple mobiles are equivalent to multiple processors. In order to reach a goal at the overall network level (the global level), all nodes need to collaborate with each other by communicating between them (interprocess communication). Due to the distributed nature and high mobility of DT-MANETs, these communications should be done asynchronously which reflects the reality. Information gathering should be done locally (using local information) in order to avoid out-of-date information. Finally, the calculation of solution should be done in a purely decentralized manner. All desirable characteristics of any algorithm used for managing topology in DT-MANETs are summarized in the following sub-section.

3.3.3.3 Desirable characteristics of algorithm for managing topology in DT-MANETs

From the obstacles and different algorithm characteristics stated above, the topology management algorithm used for DT-MANETs should have desirable characteristics as follows.

- Purely decentralized in nature.
- Considering a small input such as local information (i.e., one-hop information) and makes local decision
- Mutually coordinated entity for each entity to manage overall topology asynchronously.

3.4 Spanning Tree

3.4.1 Motivation

Definition 36 (Spanning Tree).
Given a connected graph $G_c = (V_c, E_c)$ where G_c has only one connected component (P_1). A spanning tree of the connected graph G_c is a covering subgraph which includes every vertex of V_c and which is a tree (acyclic graph). Any spanning tree must necessarily contain $|V_c| - 1$ edges. For any connected graph G_c, a spanning tree is denoted as $\gamma = (V_c, E_\gamma)$ where $E_\gamma \subseteq E_c$ and $|E_\gamma|$ equals to $|V_c| - 1$.

Given $G(t)$ a dynamic communication graph with m connected components at time t, $G(t) = \bigcup_{1..m} P_i(t)$. At time t, a spanning tree can be build upon each connected component such that there exists m spanning trees at time t of $G(t)$.

Thus, the concept of spanning tree exists only for a connected graph G_c where it is possible to include every vertex of G_c in a tree. Spanning tree has been studied and used widely in many fields such as VLSI circuit design problems [78], communication network design problem [79], and other engineering applications. In communication network, the basic task is to transport information from nodes to nodes. Therefore, an important activity in communication network is to rule or to guide the information through network. The rule for how to route the traffic through the network is based on the used routing algorithm. The very first benefit of using spanning tree as a routing algorithm falls on the fact that tree has only $n - 1$ links over a connected network. Thus, in a fully connected network, spanning tree has a benefit as a small portion of communication links is used. Apart from routing itself, spanning trees are also used in broadcasting process [80], [81], [82]. Broadcasting is a process in which a source node sends a message to all other nodes, typically found in bus topology of ethernet network and in general case of wireless network. Concerning MANETs in particular, broadcasting is important for routing information discovery, especially in reactive routing scheme where routing protocols determine routes on demand. The acyclic structure of the broadcast tree ensures that nodes broadcast message only once, and thus spend low bandwidth usage [82]. Spanning tree construction is frequently handled as an internal process of other protocols such as multicasting, clustering [83], [84], dominating set [85], generating minimum steiner tree [86], and self-stabilizing protocol [87].

In this work, we focus our studying of topology management for DT-MANETs on a logical topology approach of spanning tree. For the rest of this section, we discuss the problem of constructing spanning tree in different domains and provide accordingly state-of-the-art.

3.4.2 Constructing and Maintaining Spanning Forest in MANETs

A tree is defined as an acyclic graph. A graph composed of several trees is called, henceforth, a forest. According to the description of spanning tree shown in Definition 36, given a connected component m comprises of V vertices and E edges and two different trees, $tree_1$ and $tree_2$, where $V_m = V_{tree_1} \cup V_{tree_2}$. Then, these two different trees are not called spanning trees but spanning forest because the forest span over the connected component. The formal definition is given below:

Definition 37 (Spanning Forest).
Given a graph $G = (V, E)$*, a spanning forest* F *is a set of trees* $F = \{tree_1 = (V_1, E_1), \ldots, tree_m = (V_m, E_m)\}$ *such that* $V = \bigcup_{1\ldots m} V_j$.

In this section, literature of algorithms used for constructing and maintaining spanning tree and forest in the context of MANETs are presented.

Sridhar et al. [88] proposed a distributed algorithm for maintaining a forest of spanning trees in asynchronous communication (without global clock). The application is aimed for routing process. The drawback of the algorithm proposed in this work falls into the merging tree process that need an approval from the root node of the tree (which is used to prohibit cycles). The detailed scenario is as follow: if a node x comes in contact with a node y, then x makes a request to root node of x. This node x waits for an estimated time to get permission to merge from the root node, which can be a grant or deny message, resulting in merge or abort the process respectively. Such protocol encounters the difficulties and complexities due to the mobility of the underlying dynamic ad hoc network. Furthermore, having a root node is clearly an act of a centralized administration. Since the protocol utilized an existing central authority, we consider such algorithm as a centralized algorithm.

Jüttner et al. [82] proposed 'TreeCast' algorithm to construct and maintain spanning tree for ad hoc networks. The constructed tree will be used as a broadcasting backbone. TreeCast is a fully distributed, fully decentralized, and asynchronous algorithm. One drawback of the algorithm is in the updating new tree identification process called new $TreeID$ and followed by $NewID$ process. This process is provoked after a tree link ceases. This new $TreeID$ will be broadcasted to every nodes inside the same tree and each node will update the new treeID only if its value is higher than the current one. As a consequence, high number of maintenance message are used. Fortunately, TreeCast is meant for a static to low mobility environment found in office-like where the speed does not exceed 6 km/hrs (or 16 m/s). Due to this static and low mobility, the 'NewID' process launching after a tree link ceases does make sense. Otherwise, the updating process might fail before the information reach all nodes which will effect the process of broadcasting since logically the node think they are in different trees.

In fact, an ideal algorithm for constructing spanning tree in DT-MANETs should match the requirement stated in subsection 3.3.3.3. To the best of our knowledge, DA-GRS (a graph relabeling system) has the best potential for such requirements. We discusses about this potential and some drawbacks that needs improvements in details in the next subsection.

3.4.3 Dynamicity Aware - Graph Relabeling System (DA-GRS)

Dynamicity Aware - Graph Relabeling System (DA-GRS) [89] is an extension of Graph Relabeling System, GRS [90]. It is a high level abstraction model that can improve the development of self-organized systems. All the mechanisms within DA-GRS are suitable for managing DT-MANETs efficiently. DA-GRS models topology changes and interaction between devices following such topology changes using label system. However, it does not create services or applications itself.

Following Definition 37, DA-GRS proposes some rules for constructing and maintaining a spanning forest in DT-MANETs which are presented in Figure 3.3. In this figure, the black circle represents a node. Letters on top of the nodes mean: (1) 'T' if the node possesses the token, (2) 'N' if the node does not possess the token, and (3) 'Any' when the node can possess or not the token. The labels '0', '1' and '2' on the edge represent the route to the token. And finally, label 'off' describes a broken link.

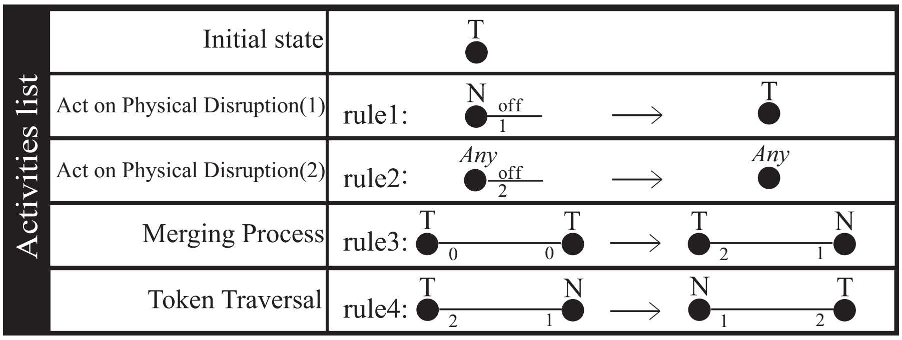


Figure 3.3: DA-GRS rules for creating and maintaining spanning forest topologies with our mapping process name

Dynamic networks are characterized by mobility and possible connection disruptions, hence, devices need to handle with these changes when creating and maintaining the spanning tree. DA-GRS proposed four rules (as shown in Figure 3.3) to handle with four different situations. In the initial state every device has the token (what means it is a tree itself), and these 4 rules are:

- rule 1: A tree link breaks, and the node belongs to the sub-tree which does not possess the token. In this case the node must regenerate the token, otherwise there will exist a tree without a token (which is an undesirable situation).
- rule 2: A tree link breaks, and the broken link occurs at a node which currently belongs to the sub-tree which possesses the token. In this case, the node does nothing regarding the maintenance of the token.
- rule 3: When a node with token meets another device possessing a token; both nodes will try to merge their trees in order to obtain a bigger tree from the two existing ones. The trees merging process starts. The result of this rule remains a bigger tree and only one token (the merging process discards one token automatically in order to remain one and only one token within a tree).
- rule 4: Token traversal in general case: the token visits the nodes of the tree following a given strategy.

An important feature of this model is that in each tree one and only one token exists. Furthermore, only two nodes possessing token (thus there exists different trees) can start the trees merging process. As we are dealing with trees, cycles are not allowed. DA-GRS manages to avoid them since it is not possible to have two nodes belonging to the same tree and possessing a token at the same time.

Regarding the 4 rules of DA-GRS's spanning forest, we map all rules into different processes as shown in Figure 3.3 and give analysis as follow. First, it uses the token management to ensure the loop free of any tree. Moreover, the node possess the token can be regarded as a root of tree and is granted to do merge operation. This means the root of tree is always walking through tree which increases the chance in order to meet with other tokens and also reduce the communication messages for asking permission and granted or denied request to merge as found in [88]. Furthermore, the algorithm described here concerns only the label changes (status changes) within one-hop neighbor. This follows the requirement from subsection 3.3.3.3, stating that only local information is available to use.

However, the synchronization of the model is not compatible to asynchronous communication as we expected. DA-GRS's spanning forest uses rendez-vous assumption as synchronization method at merging process. This rendez-vous assumption states that at one moment in time, only two nodes possessing token can meet and be merged. Furthermore, the token traversal process is based on randomness. This stochastic process makes it difficult to converge all trees into a spanning tree (i.e., slow to form a single spanning tree over a connected component). The token traversal process has been studied extensively in [91] and TABU heuristics, a deterministic based heuristic, has been proposed to improve the convergence of multiple trees over a connected component and finally becomes one spanning tree. Other important points are (1) DA-GRS model is an abstract and high level model, (2) it is incapable of communication realization and neighbor discovering method.

In summary, DA-GRS proposed a distributed algorithm that use only local information (e.g. one-hop information) to construct and maintain spanning forest over DT-MANETs. As drawbacks were found, this study improves DA-GRS to construct and maintain spanning forest in DT-MANETs. Hereinafter, we use the word 'DA-GRS reference algorithm' to represent these original 4 rules algorithm (shown in Figure 3.3) for constructing and maintaining spanning tree presented in this section, while 'DA-GRS model' is used to refer to its original meaning as a model.

3.4.4 Problems Relating to Spanning Tree

3.4.4.1 Minimum spanning tree (MST) problem

A minimum spanning tree of a weighted graph is a spanning tree of least weight, that is, a spanning tree for which the sum of the weights of all its edges is minimum among all spanning trees. The minimum spanning tree problem has been extensive studied since the first minimum-spanning-tree algorithm of Borůvka (1926), Vojtěch Jarník (1930), Kruskal's algorithm (1956) and Prim's algorithm (1957), [92], [93], [46] and [47] respectively. The formal definition of minimum spanning tree (MST) is defined as follow:

Definition 38 (Minimum Spanning Tree (MST)).
Given a connected, undirected graph $G = (V, E)$, where V is the set of vertices, E is the set of edges. A weight $w(u, v)$ is associated to each edge $(u, v) \in E$. A minimum spanning tree $T = (V, E_T)$ is a spanning tree over G such that $w(T) = \sum_{(u,v)\in T} w(u, v)$ is minimized.

Algorithms for Optimal MST The most famous algorithms for solving this MST problem are Kruskal's algorithm and Prim's algorithm. In Kruskal's algorithm, the set A is a forest. The safe edge added to A is always a least-weight edge. On the other hand, in Prim's algorithm, the set A is a tree, and the growing edge only added to the formed tree. The two algorithms are greedy algorithms and solve MST problem in polynomial time. Basically, at each step of algorithm, one of several possible choices must be made. The greedy strategy advocates making the choice that is the best at the moment. Based on static and a connected graph, moreover, the global information is assumed (i.e., number of edge, node and weight value of each edge are known). We consider this case is an easy problem of finding minimum spanning tree and it has been proven in literature that greedy strategies do yield a spanning tree with minimum weight when global information is provided [94].

In fact, apart from the mentioned algorithms, Graham et al. [95] reviews the history of MST algorithms and shows other algorithms for finding optimal MST, such as a depth-first search approach [96].

Distributed Algorithm for MST The distributed or parallel algorithm for MST problem has been studying since [97] (1983) and still is interesting in many research fields, main examples are of parallel computing and networking (e.g., overlay network, MANETs). Michalis et al. [98] gave an extensive study on literature for optimal distributed algorithm for MST problem in terms of time and message complexity. The most common distributed algorithm is the Spanning Tree Protocol (RFC 1493) [79], used by OSI link layer devices to create a spanning tree using the existing links as the source graph in order to avoid broadcast storms in classical networks. The algorithms found in both [97] and [98] are distributed but only consider a fixed network infrastructure. Furthermore, all algorithms above assume unlimited bandwidth and no collision of messages which are found in DT-MANETs. Meanwhile, some works focus at constructing MST within a bound communication rounds and number of messages such as [99]. However, it, again, considers no boundary level of local information and thus every node will receive fragmented local information from all nodes in the network and then each node can compute global information.

According to our previous discussion in 3.3.3.3, the mentioned algorithms above are not suitable for managing minimum spanning tree over DT-MANETs. In summary, they are distributed and work in asynchronous mode, but they are neither considering purely local information input nor limited memory and computing resources.

3.4.4.2 The multi-criteria MST problem

Considering a weighted graph $G = (V, E, w^k)$ where w_e^k denotes the weight of edge $e \in E$ at particular weight set k, where $k = \{1, 2, \ldots, m\}$ denotes different set of weight which may represent the distance, cost and so on. The multi-criteria MST problem (mc-MST) can be formulated as follows:

Definition 39 (mc-MST problem)**.**

$$minf_1(\gamma) = \sum_{e \in E_\gamma} w_e^1;$$

$$minf_2(\gamma) = \sum_{e \in E_\gamma} w_e^2;$$

$$\dots$$

$$minf_m(\gamma) = \sum_{e \in E_\gamma} w_e^m;$$

where $f_i(\gamma)$ is the ith objective to be minimized for the problem

Guolong et al. [100] claimed that the multi criteria minimum spanning tree (mc-MST) is a NP-hard problem, although the basic MST problem is solvable in polynomial time. Accordingly, approximation methods must be used if we want to tackle it efficiently. Shown in [100], an improved enumeration algorithm to give out all real Pareto optimal solutions for the mc-MST problem is presenting. However, the mc-MST problem of [100] considers a connected and undirected graph.

3.5 Robust and Efficient Topology Management in DT-MANETs

Definition 40 (Robust - Cambridge Advanced Learner's Dictionary).
strong and unlikely to break or fail

Definition 41 (Efficient - Cambridge Advanced Learner's Dictionary).
working or operating quickly and effectively in an organized way.

Let take the example of road system, the basic desire of robust and efficient topology of such system are:

1. Having the amount of traffic at an acceptable level (i.e., not too much of traffic jam)
2. When one road is blocked due to accident or routine maintenance, there is at least another route as an alternative route
3. Road system spans and can cover maximum area.
4. Having high quality and safety of the road system (i.e. road's surface condition, properly giving warning sign, etc.)

Comparing those basic desired points with the networking topology, the first case is as providing a big enough *bandwidth*, the second is as availability of *the second routing path*, the forth is about *coverage area*, and the latter case is about *the quality of communication path.*

Scientifically, we would like to quantify how robust the system or object does. Ali et al. [101] tried to answer that and developed the following three questions as a methodology to quantify the robustness of a resource allocation system.

- What behavior of the system makes it robust?
- What uncertainties is the system robust against?
- How exactly is the system qualitatively robust?

Referring to the quantifying robustness methodology above, we can summarize that robustness characteristics of a system are defined by a set of behaviors and uncertainties of the system. This implies that the robustness characteristic of a system may takes into account a single or multiple quality criteria to enhance robustness of the system against a set of particular behaviors and uncertainties. This study follows the above suggestion methodology and applies them to define robust topology management in DT-MANETs context. In a dynamic network like DT-MANETs, most of entities are usually mobile. Thus, the meeting and departing of nodes are taken as a natural behavior and not a failure behavior. Accordingly, the partitioning of a network into multiple sub-networks or subgraphs is a common behavior. Unfortunately, this common behavior makes the end-to-end communication in such network delay or disrupt which deteriorates the robustness of system. In summary, the appearance and disappearance of the network entities effects the uncertainty of end-to-end communication success in DT-MANETs. Keeping such natural behavior of DT-MANETs in mind, there are two approaches to achieve a secure or robust topology system. First, the managing units must seek a robust topology such that avoiding as much as possible these kind of natural behavior in order to provide a better quality of service. Second, the managing unit must provide an alternative topology such that even one path disrupt, the alternative path can operate instead. Finally, the system becomes robust and then providing a better quality of service. Certainly, a robust system can operate both approaches at the same time. This study focuses on the former approach; that is selecting a topology that avoid the deterioration of network or increase the quality of service in DT-MANETs by utilizing different quality factors existing in each entity of a particular communication network $G = \{V, E\}$. In Chapter 4, we defined and discussed on these different quality factors.

As discussed earlier, managing topology in DT-MANETs should use decentralized, asynchronous and using local knowledge algorithms. This means the decision need to be made by individual node utilizing local information (information from neighbors or local structure) since global information is impossible or not practical to obtain. However, this locally individual decision has to enhance the robustness quality of overall topology structure. The robustness quality, then, concerns three different levels:

1. quality of virtual backbone at the communication graph level,
2. quality of the topological structure itself (i.e. spanning tree, dominating set, cluster),
3. quality of the individual entities (i.e. node's quality and edge's quality).

Quality examples of virtual backbone or managed topology T at the communication graph level are, such as, the coverage area criterion, the back up path (second routing path) and how fast the virtual backbone covering area, etc. For quality of topological structure, the quality is subjective to the structure and the specified problem. For examples, clustering problem tends to find the minimum size of cluster head in a network or to restrict number of nodes under one cluster head. On the other hand, in connected dominating set problem, minimizing number of backbone nodes is as crucial as providing the backing up route for any node in the network. Furthermore, for both clustering and connected dominating set techniques, quality of nodes, such as energy availability. is also used for selecting high quality node as cluster-head or dominating nodes. For spanning tree problem, loop is prohibited and the high quality of crucial communication link should be selected as tree edge, etc.

As seen from above examples, the quality of individual entity also influences the quality of topology structure also. Nodes and edges are the main elements in every networks. Each of these elements has their own properties and a given measurement characteristics. It is interesting to discuss extensively on different characteristics that influence the robustness of the topology management and are suitable to be used in DT-MANETs context. This topic are discussed extensively under the next Chapter.

In managing robust topology graph $T = (V_T, E_T)$ such that $V_T \subseteq V$ and $E_T \subseteq E$, a proper set of edges and nodes must be selected in order to achieve the desired quality topology graph. At a particular moment and a configuration of network, the robust topology may have different meaning and needs. Similar to other real world applications, robust topology may comprises of many factors of network elements' qualities. For some qualities inside elements, they may have a strong correlation or no correlation. For the latter case, it may happen that they are conflicting qualities. This work is interested in managing robust topology in DT-MANETs context based on conflicting criteria (non-correlation criteria) of nodes and edges.

Definition 42 (Conflicting objectives).
The objectives are in conflict with each other if an improvement in one objective leads to deterioration in another.

Thus, the quality of nodes and edges promise to form the robustness metric for managing topology. So far, we discussed briefly on examples of efficiency and quality of entities metrics for being a robust topology in DT-MANETs. Chapter 4 discusses this in depth. Obviously, finding a robust topology is a multi-objectives problem, as defined in Definition 20. If the two qualities are conflicting with each other, whenever one quality is met, another quality of the solution may decrease. Finding the best robust topology solution from those multiple objectives are done by means of optimization.

Indeed, finding the best robust topology solution (see definition of Pareto front in 24) from multiple objectives can be given by means of optimization. More specifically, it is done by multi-objective optimizer. Ideally, the optimizer is granted the global information of network topology. It calculates the best result and disseminate this best result to network nodes such that each node configures itself accordingly. However, we discussed previously under 3.3.3.3 that DT-MANETs should manage topology using distributed algorithm, considering local information and communicating asynchronously. Hence, the ideal process becomes infeasible in the environment of DT-MANETs. The practicing approach of researchers and practitioners is to have distributed algorithms with some heuristics to manage the desired topology. However, the only way to confirm the performance of the proposed algorithms is to compare with the best solutions illustrated by the best Pareto front shown in literature.

3.6 Summary

This chapter introduces the definitions of topology and topology management, and illustrates some examples of topology which can be found in MANETs. The motivation of topology management mainly due to many uncertainties and obstacles found in the network, and thus, topology management aims to enhance the communication quality at different aspects by different techniques. There exists two main aspects of topology management. The first aspect deals with physical topology directly. In this aspect, the topology algorithms change the existing topology by controlling power consumption at different functions of network nodes, such as reducing transmitting power, shut down some particular node for saving energy. The second aspect, which we are highly interested in, aims at constructing a managed topology subgraph logically (application level), sometimes is called virtual backbone. In this second aspect, research works in literature can be divided into two sub-categories, which are hierarchical network topology and flat network topology. Dominating set and clustering techniques are used to generate a so called hierarchical network where dominating set and clusterhead acting like routers and have more controlling functioning at hand. In contrary, spanning tree and spanning forest are used in flat network topology scheme which every node has the same functionality.

Centralized algorithms is suitable for managing topology in a static and small network. However, managing topology in DT-MANETs, with high mobility node, needs distributed algorithms with asynchronous communication protocol. More specific, we are interested in a distributed algorithm that use only one-hop information and makes locally decision for constructing and maintaining a virtual backbone on top of existing networks.

Spanning tree is an important structure to many applications especially in networking field. The beneficial characteristics of spanning tree also draws attentions of researchers in the field of ad hoc networks. An algorithm proposed by DA-GRS model meets some characteristics of being a good algorithm for managing topology in DT-MANETs. Robust topology is characterized by the desire to overcome some adverse behavior of the network. A virtual backbone or managed topology should provide robustness at both communication graph and topological structure level. This study aims at providing robust topology management in DT-MANETs by considering different robustness qualities found in nodes and edges. These robustness qualities of nodes and edges are discussed extensively in the next Chapter. Robustness is subjective to a specific system. Thus, robustness problems may concern multiple criteria at the same time. Since these multiple criteria may be in conflict, multi-objective optimization is a technique which may provide the best set of solutions.

Chapter 4

An Efficient and Robust Topology: Spanning Forest Approach

Contents

4.1 Introduction 52

4.2 Efficient Spanning Forest for Topology Management 52

4.2.1 Motivation 52

4.2.2 Efficiency Metrics 53

4.2.2.1 spanningRatio() function 53

4.2.2.2 spanningSpeedRate() function 54

4.2.3 Utilizing and Applying Spanning Forest Algorithm of DA-GRS . . . 54

4.2.3.1 Communicating Operation 54

4.2.3.2 Theoretical Performance 56

4.2.4 Efficient Spanning Forest by means of Tree Traversal Strategies . . . 58

4.2.4.1 Introduction 58

4.2.4.2 State-of-the-Art 58

4.2.4.3 Proposition of a Solution: 'DFSmove' heuristic 60

4.3 Robustness Quality Criteria 62

4.3.1 Cooperative Enforcement Aspect 62

4.3.2 Energy Availability Aspect 63

4.3.3 Capacity Bandwidth Aspect 68

4.3.4 Discussion on other criteria 69

4.3.4.1 Signal Strength 69

4.3.4.2 Age of edges 70

4.4 Robustness Metrics 71

4.4.1 nodeWeight() function 71

4.4.2 isolateLowQualityNode() function 72

4.4.3 pathWeight() function . . . 73
4.5 Robust Spanning Forest by Nodes and Edges' qualities . . . 75
4.5.1 Problem Description: Cooperative Enforcement Aspect . . . 75
4.5.2 Problem Description: Energy Availability Aspect . . . 76
4.5.3 Problem Description: Capacity Bandwidth Aspect . . . 77
4.6 Solutions for Robust Spanning Forest in DT-MANETs . . . 78
4.6.1 Introduction . . . 78
4.6.2 Construction and Maintenance Process . . . 79
4.6.2.1 G-NODE . . . 79
4.6.2.2 G-PATH . . . 79
4.6.3 Re-organization Process . . . 80
4.6.3.1 BREAK heuristic . . . 81
4.6.3.2 CHANCE heuristic . . . 81
4.7 Multi-objectives Approach for Managing Topology in DT-MANETs 85
4.7.1 Problem Description . . . 86
4.7.2 Proposition of a solution: 'G-Node-Path' . . . 86
4.7.3 Optimum solution to single-objective function . . . 88
4.7.3.1 Reduction of $nodeWeight(\gamma)$ objective . . . 88
4.7.3.2 Reduction of $pathWeight(\gamma)$ objective . . . 90
4.7.4 Optimal solution to multi-objective function . . . 92
4.7.4.1 Enumerating the whole set of spanning trees: Exact Approach 92
4.7.4.2 Meta-Heuristics Approach . . . 93
4.8 Summary . . . 94

4.1 Introduction

In this chapter, most of contributions are presented. They comprise of motivations, problem descriptions, cost functions (efficiency and robustness metrics) and proposal of the solutions. Since the work has many facets to tackle, each contribution has been categorized under different sections. First, the efficient spanning forest for topology management is discussed. The efficiency metrics are given as spanningRatio() and spanningSpeedRate() functions. Since DA-GRS model is 'only' a graph and labeling model, the instantiations of this model, including the token traversal strategy and the communication protocols, are needed in order to have a real implementation and the corresponding evaluation.

The characteristic of robust spanning forest topology and the robust qualities are discussed. The robust spanning forest can be elaborated throughout different approaches. This study focuses on cooperative enforcement aspect, energy availability aspect and maximization of minimum capacity bandwidth aspect. These approaches characterize the robustness of topology especially in DT-MANET. The robustness metrics are given as weight(), isolateLowQualityNode() and pathWeight() functions. These metrics evaluate the robustness of spanning forest structure in a different contexts. In fact, the robust topology problem is more suitable to address as a multi-objective problem where multiple quality criteria are taking into account at the same time. We address both single-objective problem and multi-objective problem with proposal of solutions along with the obtaining process of optimal solution(s) for evaluation purpose.

The algorithms and heuristics proposed to acquire the topology according to the discussed robust characteristics are provided. Each algorithm and heuristic strictly follow the desired characteristic discussed in section 3.3.3.3 which are being distributed, using local information (one-hop information), making decision locally and working in asynchronous manner.

4.2 Efficient Spanning Forest for Topology Management

4.2.1 Motivation

As discussed earlier in section 3.3.2.2, the main properties of managed topology subgraph T (virtual backbone) concerns particular characteristics such as being connected and having low stretch factor. We discussed a bit further in section 3.5 about robust topology management that coverage area of the virtual backbone T and how fast T can cover the area are also important factors. These characteristics give direct influence to the efficiency of the virtual backbone T at the communication graph level.

In this work, we study how to have an efficient distributed algorithm to provide efficient spanning forest topology in the context of DT-MANETs. Since the DT-MANETs are very dynamic network, our concerns are described as follows:

- How does a tree structure span over a connected component of communication graph(and become spanning tree)?
- How fast can the tree span to cover a connected component of communication graph?

We proposed two efficiency metrics according to our concerns above and are presented in the next section 4.2.2.

As discussed in section 3.4.3, DA-GRS algorithm is a promising decentralized algorithm for constructing spanning forest. However, it has some weakness which can be improved further for working with DT-MANETs. The very first to consider regarding the spanning forest algorithm is the communication protocol. This algorithm is based on DA-GRS model which only concerns graph relabeling system and not provide any underlying communication mechanism. In order to utilize spanning forest algorithm over a DT-MANET, nodes must exchange some messages between them. These messages are used as a basic mechanism to gain knowledge about existing neighbors of a node at one moment. Since no global knowledge is considered, a more detailed communication syntax needs to be specified and organized for different events of the algorithm. Therefore, the proposed message sequence that devices must exchange in a decentralized system is explained in the section 4.2.3. Finally, our proposed 'DFSmove' heuristic for efficiency improvement of DA-GRS algorithm is discussed in section 4.2.4.3.

4.2.2 Efficiency Metrics

We propose two different efficiency metrics, spanningRatio() and spanningSpeedRate() function. The first metric has been used in several works such as [72], [91]. On the other hand we introduced the second metric for the first time which is also related to our experimentation methodology. Both efficiency metrics are described as the following.

4.2.2.1 spanningRatio() function

At a given moment t, $G(t)$ may be partitioned into a set of m connected subgraphs. Having Γ as the set of all trees at moment t of $G(t)$. The quality of topology can be assessed by the number of connected subgraphs (m) over the number of trees created (Γ). This quality is determined by the following ratio.

$$spanningRatio(G(t)) = \left(\frac{m}{\mid \Gamma \mid} \right) \tag{4.1}$$

The value of the performance ratio approaching to one means higher quality of the topology (less trees in a connected subgraph). Having a spanning tree per connected subgraph enables more efficient communication and topology management, since at least, the information can be disseminated systematically via the created spanning tree. This means the algorithm is robust against the dynamism of the network because it can construct a spanning tree covering all the nodes belonging to the connected subgraph.

Figures 4.1(a) and (b) illustrate the measurement of efficiency functions proposed here. In the figure 4.1(a), the communication graph $I(t)$ has two connected subgraphs, and each connected subgraph has one spanning tree. On the contrary, the communication graph $K(t)$ depicted in figure 4.1(b) has only one connected subgraph but four spanning trees ($\gamma_1, ..., \gamma_4$). Thus, the $spanningRatio(I(t))$ and $(K(t))$ equal to 1 and 0.25, respectively.

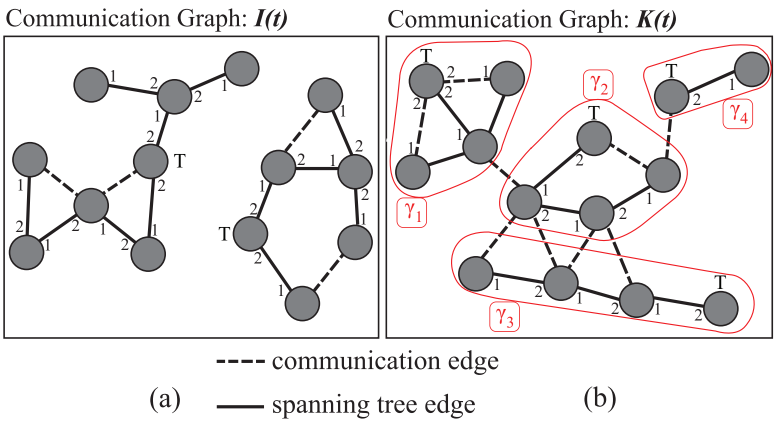


Figure 4.1: An example scenario for illustrating the proposed efficiency metrics

4.2.2.2 spanningSpeedRate() function

To answer the question of how fast trees merge into a spanning tree, we proposed spanningSpeedRate() function. The *spanningSpeedRate*() is measured based on the number of iterations per simulation. Let Δ be the number of iterations the algorithm required trying to achieve the least *spanningRatio*() and Δ^* be the number of iterations required per $G(t)$. Having *spanningRatio*() equal to one within $G(t)$ is an ideal situation. However, having limited merging process (will be explained in sub-section 5.1.3.2) causes no guarantee that *spanningRatio*() will be one, in other words, it is always possible to have multiple trees per connected component at any time t of communication graph G. In such case, the number of iterations used within that $G(t)$ will be counted into Δ. The lower the value of *spanningSpeedRate*() is, the faster the algorithm converges a connected component into a spanning tree. The *spanningSpeedRate*() can be written as below.

$$spanningSpeedRate(G(t)) = \left(\frac{\Delta(G(t))}{\Delta^*(G(t))}\right) * 100 \tag{4.2}$$

4.2.3 Utilizing and Applying Spanning Forest Algorithm of DA-GRS

4.2.3.1 Communicating Operation

[IAIT09] proposed the details of communication sequence for the spanning forest algorithm of DA-GRS as follows:

- **Beaconing:** In order to have knowledge of the one-hop neighborhood most decentralized systems utilize beacons (also called 'hello messages') [102]. For that purpose, every node sends periodically a message alerting about its presence. For considering a node as a neighbor, one must receive a beacon of the node regularly. A node will not be a neighbor anymore when its beacon is not received within a predefined time.

Using this beaconing both a broken communication link and the appearance of a new one-hop neighbor are detected, and thus, 'rule 1' and 'rule 2' in Figure 4.2 can be applied. Based on *Beaconing Rate* of IEEE802.11, Local Area Network, [103], the time interval used for periodically sending the beacon is 100 millisecond.

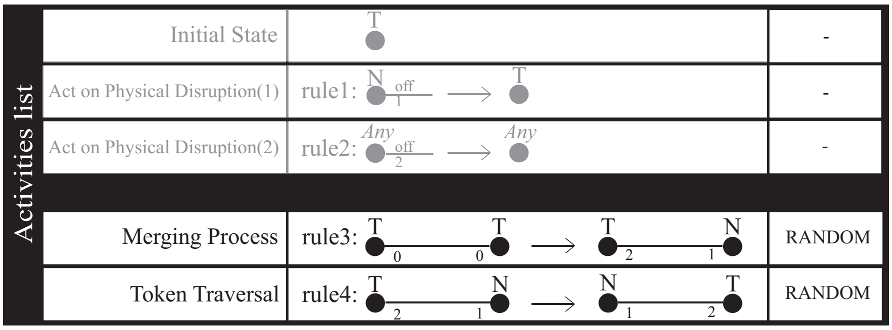


Figure 4.2: Heuristics used in DA-GRS as token traversal strategy and Merging Process Strategy

- **Trees Merging Process** 'rule 3' in Figure 4.2 represents the spanning tree construction scenario (trees merging process). DA-GRS uses rendez-vous assumption [104] as synchronization method at this merging process. This rendez-vous assumption states that at one moment in time, only two nodes possessing token can meet and be merged. We consider that this assumption is too rigid in real world communications. Thus, this work proposes to relax this assumption by allowing a node to choose one token among the tokens owned by its neighbors.

  In a distributed system a node has no ability to know if there exists any node with token in its neighborhood. Thus, nodes holding a token will broadcast a packet, *'findingTk'*, to verify whether any of its neighbors also possesses a token. If any neighbor of this broadcasting node possesses a token and receives *'findingTk'* will reply using a *'ACK_finding'* message. *'ACK_finding'* is an expression of agreement to merge their trees.

  Moreover, this particular neighbor will set its status to wait for *'SYN/ACK_finding'* message to confirm the merging process within a predefined period, *'TimerWaitFor_Syn AckFinding'*. As we are working with a discrete simulator, the time duration of the timers is one simulation step.

  After broadcasting *'findingTk'*, the broadcasting node will wait within a predefined duration, *'TimerWaitFor_Finding'*. At the end of this waiting time, the broadcasting node selects one of its neighbor and a *'SYN/ACK_finding'* message will be sent using unicast to this selected neighbor. In case, there is no node with token in the neighborhood, at the end of this timer the token is circulated. The message sequence of this process is illustrated in Figure 4.3.

- **Token Traversal** 'rule 4' in Figure 4.2 stands for token traversal in general case. When a node sends a broadcast message for finding a neighbor possessing a token, it also establishes a timer as addressed in previous section, *'TimerWaitFor_Finding'*. If

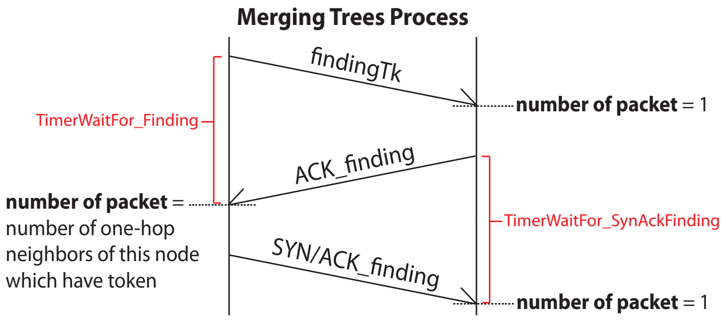


Figure 4.3: Message sequence diagram for merging trees

the timer finishes and there is no answer from any neighbor, the token movement takes place. If there is no neighbor belonging to a different tree, the node will directly move the token, see Figure 4.4.

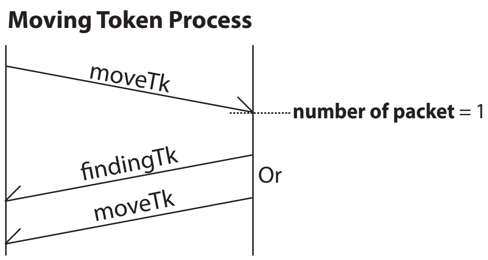


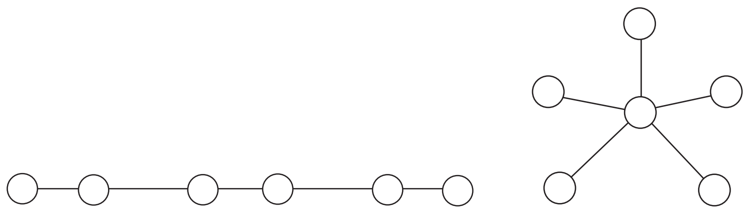
Figure 4.4: Message sequence diagram for traversing the token

Please note that the selection process in both merging process and the token traversal is done randomly in DA-GRS algorithm. In summary, our proposal adhere to the initial state and rule 1 and 2 of DA-GRS algorithm. However, for rule 3 and 4 (the merging process and token traversal) have been modified and add on details of communication in this study. Actually, these two processes are the main promising processes for improving DA-GRS algorithm to become a distributed and localized algorithm for TM. Figure 4.2 emphasizes the fact that this study focuses on these two processes in particular and also specifies the heuristic used by DA-GRS for both Merging Process and Token Traversal activities.

4.2.3.2 Theoretical Performance

In this section we briefly discuss the theoretical performances of the DA-GRS reference algorithm with our details of communication sequence. Let us consider a communication graph $G = (V, E)$. The node who possesses token sends a short message and waits for its neighbors to reply within a constant time. Sending and receiving message within one hop neighbor has an $O(1)$ time complexity. Let us consider two different scenarios shown in Figure 4.5.

In Sub-Figure 4.5(a), considering a linear topology, the best case scenario when multiple nodes out of communication range of each other can communicate at the same time, the algorithm needs $(log(|V|))$ steps to be achieved, each node only dealing with one or two neighbors, the complete time complexity is then $O(log(|V|))$. While in Sub-Figure 4.5(b)



(a) Example of linear topology (b) Example of star topology

Figure 4.5: Example topology used for complexity explanation

depicts the worst case scenario of star topology that $(|V|-1)$ nodes connect through a single node. Since each time the tree link is constructed, it has a time complexity of $O(1)$. Considering the entire graph, the algorithm reaches its objective in $|V|-1$ steps, and thus the time complexity is $O(|V|-1)$. Considering dynamic network and also the collision of messages that can occur, we cannot provide a precise theoretical complexity. In this case, we can only conclude that the DA-GRS reference algorithm has a time complexity of between $O(log(|V|))$ and $O(|V|-1)$.

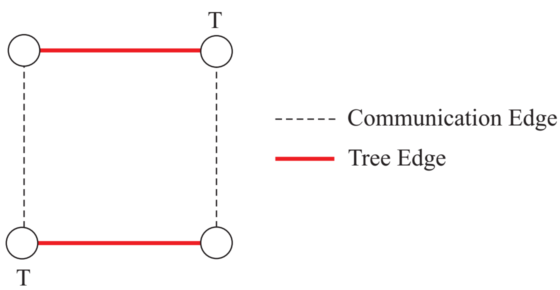


Figure 4.6: An example of no convergence case using DA-GRS reference algorithm

Let us consider a message complexity of DA-GRS reference algorithm in theoretical point of view. If the communication is in synchronous manner (global clock is used for synchronizing communication), there exists at least a case that the algorithm cannot converge. It is worth to emphasize that this situation is found only in static network which uses synchronous communication and is shown in Figure 4.6. Since this study considers dynamic network and asynchronous communication, such blockage situation is not realistic. However, we cannot provide a precise theoretical message complexity by the same reason (considering dynamic network). Pigné et al. [105] observed simulation results and concluded that the DA-GRS reference algorithm need an exponential messages to converge the spanning tree in static network.

4.2.4 Efficient Spanning Forest by means of Tree Traversal Strategies

4.2.4.1 Introduction

One common mechanism used in spanning tree algorithms is the utilization of tokens [106], [107]. In [108], the authors state that techniques for traversing the token that perform well in static networks are not necessarily well suited in networks with high mobility. Thus, a new study of token traversal in high mobility network must be undertaken. Also in [91], it is concluded that the token movement strategies impact on the tree construction, and, therefore, on the topology management both in terms of speed and performance. This motivated us to study, implement and compare different token traversal techniques in order to determine which strategy performs better in different environments in DT-MANETs. Up to now, the token traversal strategies used in DA-GRS are based on the assumption that no memory is used in the mobile nodes. These techniques are random and Tabu [91].

In this study, we applied Depth First Search (DFS) for the first time to DA-GRS. We considered to include DFS in our comparison since it is a very well known strategy for static tree traversal and the idea of DFS has been used for mobile ad hoc networks in recent works [80]. However, due to the highly fluctuant topology, having an ordering strategy might not be a good idea. Thus, a deep study and also a comparison between DFS and other techniques are needed.

This study assumes that spanning trees provide a reliable path for efficient communications and services. Thus, having a spanning tree covering as many nodes as possible in the shortest time is desired. In the context of this study, the spanning tree must span the entire connected communication graph. Therefore, we implemented and compared different strategies for traversing the token in the tree topology in terms of the performance ratio and the convergence speed rate. The performance ratio is measured as the number of different partitions (or connected components) of the underlying network divided by the number of existing trees. The convergence speed rate shows how fast multiple trees belonging to the same partition merge into one tree. We compare three different distributed strategies: RANDOMmove, TABUmove, and DFSmove, described later in sub-section 4.2.4.2 and 4.2.4.3.

4.2.4.2 State-of-the-Art

As explained above, in DA-GRS algorithm and usually when dealing with spanning trees, the system needs a token for creating and maintaining a tree. Every node, at some moment, must possess the token, since it allows looking for neighbors with token to merge trees. The way this token moves along the tree impacts on the spanning tree construction. In literature, tree traversal refers to the process of visiting each node in a tree data structure in a particular manner [109]. In the context of this study, we want the token to traverse less but has more chance to meet another token. In other words, we want the fastest rate of the tree construction to cover a connected subgraph, which means less number of trees or in the best case, remaining only one tree over a connected subgraph.

This section gives a detailed explanation of the two strategies found in literature, Randomness and Tabu. It is worth noting that all strategies are working in distributed, asynchronous and localized manner suiting to work in DT-MANETs.

Randomness: The 'RANDOMmove' here do random chance by following the uniform distribution law. RANDOMmove is the heuristic used by DA-GRS algorithm by default. The process is done by selecting a node randomly among the list of neighbors. The description of the 'RANDOMmove' traversal technique is described in Algorithm 1.

Algorithm 1 Using RANDOMmove heuristic in $Move\_Token$ (τ_i) process of a node ν

```
α is the set of neighbors of node ν
node ρ is a node selected randomly from set α
move token τ_i from node ν to node ρ
```

Tabu List: 'TABUmove' creates a list of forbidden movements in which the most recent nodes possessing the token are stored. This list is called tabu list. The algorithm consults the tabu list before sending the token to a neighbor in order to avoid visiting the same node repeatedly. TABUmove uses a fix size of memory, *memory_size*, to set the number of stored nodes in the list. This list is sent within the token, no node memory is used. In Algorithm 2 a detailed description of this strategy is given.

Algorithm 2 Using TABUmove heuristic in $Move\_Token$ (τ_i) using a defined value of *memory_size* processing at a node ν

```
α is the set of neighbors of node ν
β is the TABUmove list which has size equal to memory_size
Set availableNode = α - β
if availableNode ≠ ∅ then
   node ρ is a node selected randomly from set availableNode
   token τ_i move from node ν to node ρ
   if the number of item of β reach the memory_size then
      remove the first item from list β
      add ν to the end of list β
   else
      add ν to the end of list β
   end if
else
   node ρ is a node selected randomly from set α
   remove item ρ from list β
   token τ_i move from node ν to node ρ
   add ν to the end of list β
end if
```

Applying this technique to DA-GRS reference algorithm was proposed by [91]. The *memory_size* of the list (its length) was also studied in [91], and it was demonstrated that a tabu list longer than 1 entry of device did not provide much better results than using a tabu list with size 1. Therefore, we use in our study TABUmove with size list equal to 1. For brevity, henceforth we will use 'TABUmove' to represent the usage of TABUmove at *'memory_size'* equals to one. This is equivalent to prohibiting sending the token to the node from which the current one received it.

4.2.4.3 Proposition of a Solution: 'DFSmove' heuristic

Depth first strategy is commonly used as token movement technique [81, 110, 111] when dealing with tree based topologies. DFSmove imitates the traversal of the classical Depth First Search algorithm and, thus, it is an ordering traversal strategy (deterministic strategy).

In order to traverse systematically like the classical algorithm in distributed and dynamic systems, DFSmove utilizes the neighbor list information provided by the beaconing process. Thus, the neighbor list is always up to date. Furthermore, in this implementation, it is necessary to keep information inside each node. To be more specific, these information are: (a) about the node that sends the token to the current device for the first time (henceforth, we refer to this first node as 'upper neighbor'), and (b) information of neighbors receiving the token from this current device. In this way, the node will definitely sends the token to all its neighbors using the neighbor list and the information stored (a) and (b). It will not send the token back to the upper neighbor meanwhile all the list of neighbors is not visited.

The mechanism is as follows: whenever the current node receives the token back from its neighbors (and this is not the first time this node receives token), the current node will send the token to the next neighbor in the neighbor list. Once the list is finished, the token is sent back to the 'upper neighbor' if it has not gone from the neighborhood. Otherwise, this current node will become its own 'upper neighbor' and will send again the token to the first neighbor of its neighbor list. This implementation is described in Algorithm 3.

The application of DFSmove is an original contribution of this thesis. Indeed in the literature, only tabu and random movements have been considered.

Algorithm 3 Using DFSmove heuristic in $Move\_Token$ (τ_i) process of a node ν

α is the set of neighborhood of node ν
β is the DFSmove list in node ν
ϖ is '*upper neighbor*'
δ is the latest node that send τ_i to ν
if ϖ is empty **then**
    $\varpi = \delta$
end if
Set $availableNode = \alpha$ - β - ϖ
if $availableNode \neq \emptyset$ **then**
    node ρ is the first node from set $availableNode$
    move token τ_i from node ν to node ρ
    add ρ to the end of list β
else
    clear list β
    if ϖ is in the set α **then**
        move token τ_i from node ν to node ϖ
        set ϖ to empty
    else
        $\varpi = \nu$
        Set $availableNode = \alpha$ - δ
        node ρ is the first node from set $availableNode$
        move token τ_i from node ν to node ρ
        add ρ to the end of list β
    end if
end if

4.3 Robustness Quality Criteria

Quality of node and edge is one main issue used for managing topology. The quality of node and edge also influences the quality and robustness of topology structure. This section provides the discussion of a number of quality criteria found in both nodes and edges. These quality aspects are trust level of cooperative enforcement approach, battery percentage of energy availability aspect and capacity bandwidth. Moreover, some interesting but not suitable criteria are also discussed under this section too.

4.3.1 Cooperative Enforcement Aspect

Motivation Ad hoc networks rely on 'cooperation' of a set of nodes in order to emerge and operate the network. A single node can deteriorate the well-being (robustness) of an ad hoc network by not forwarding messages. There exists many reasons for non-cooperative behavior of a node. As has been reviewed in Section 2.1, typically mobile nodes have limited computing, memory and battery power. Depending on policy configured inside the node, any action that does not belong to itself maybe discarded when any resources reach a threshold.

Cooperative enforcement approaches have been proposed by a number of researchers to enhance the robustness, the availability and/or the overall throughput [112] in pure Mobile Ad Hoc Networks. The main objective of those works is to cope with 'selfish nodes'. Such nodes can deteriorate the robustness of the network because they do not give collaborative efforts (e.g., by not forwarding a packet received from others). In this paradigm, the terms of 'trust' and 'reputation' are used as judgement values representing the cooperative level or trust level toward other stations in the community. This study focuses on utilizing this trust value for strengthening the robustness of topology in a DT-MANET.

Please note that, the evaluation of trust and the dissemination of reputation system are out of concern to the scope of this study. We utilize the existing trust value on each node and only focus on robust topology of spanning forest in DT-MANETs.

Trust and Reputation Trust and reputation have a close relationship and often have been treated synonymously [113]. However, they actually have different meaning and they are depending on the context area. Trust has been studied and mentioned most in sociology, psychology and philosophy [114]. While, reputation has been explored in social and biological literatures [115]. It can be concluded that notions of trust and reputation are context dependent. For simplicity, we follow the definition of trust given by [116] as follow:

Definition 43 (TRUST).
Trust can be viewed as the expectation or the belief that a party will act kindly and cooperatively with the trusting party. [116]

Definition 44 (REPUTATION - CAMBRIDGE ADVANCED LEARNER'S DICTIONARY).

The opinion that people in general have about someone or something, or how much respect or admiration someone or something receives, based on past behavior or character.

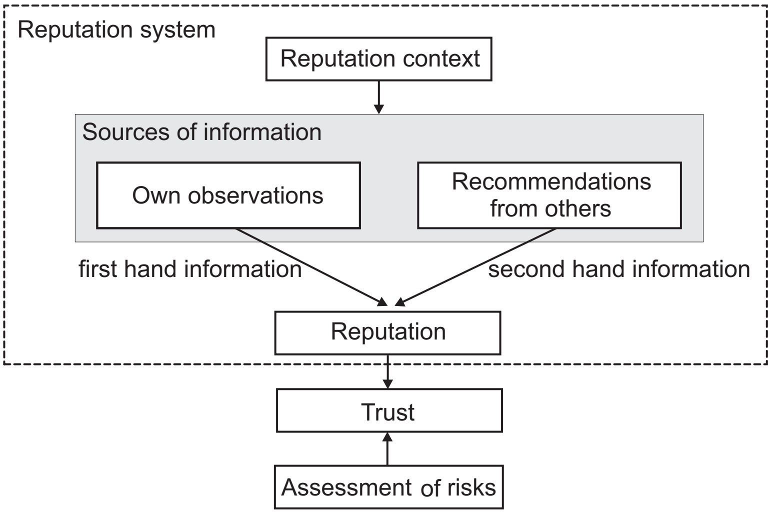


Figure 4.7: Relation between trust and reputation

The relationship between trust and reputation has been summarized in [117] as shown in Figure 4.7. From this point onward, we regard 'trust' as the resulting combination of value from reputation system and self-evaluation of risk according to the Figure 4.7. The evaluation of trust in context of MANETs has been proposed in many works, good examples of such works can be found in [116], [118], [119], etc. A comprehensive survey on cooperation enforcement in MANETs can be found in [112], while detailed discussion on peer-to-peer key and trust management approach in MANETs can be found in [120].

4.3.2 Energy Availability Aspect

Motivation Since the network lifespan strongly depends on the cooperation among mobile elements, the ability of mobile node to operate further in time is one of the most important characteristics in ad hoc network. In this section, the remaining battery of a node becomes the criteria to maintain robust topology in DT-MANETs.

Energy is one of the primary concerns when dealing with ad hoc networks. Every mobile nodes are equipped with battery at different capacity depending on the functionality of each mobile node. Specification of batteries has a vast variety of differences. In general, we found a wide range of type and capacity of batteries (i.e., NiMH, Lithium-Ion, Lithium-Ion-Polymers etc.) for different mobile nodes (i.e. mobile phone, PDA, laptop, NetBook, tablet, etc.). A typical approach when dealing with energy consumption in MANETs is to focus attention on wireless transceiver power consumption only. However, mobile nodes consume power for other operations too (i.e., multimedia players, word processor, calendar, other applications). In this problem, we consider the energy problem at a bigger scale by considering the whole battery capacity available as a criteria for finding a robust topology in DT-MANETs. This study provides a short description of battery and consumption terminology. Furthermore, the typical batteries used in mobile nodes are discussed in Sub-Section 4.3.2.

Battery and Energy: Main Concept

A cell: A cell is an electro-chemical device capable of supplying the energy that results from an internal chemical reaction to an external electric circuit.

A battery: A battery is composed of one or more cells, either parallel or series connected to obtain a required current/voltage capability (batteries comprised of series connected cells are by far the most common).

Battery Voltage (Volts): Battery Voltage is the electric potential of a battery pack. For a Lithium-polymers (Li-Po) battery pack, the voltage can be estimated only by knowing the number of cells inside Li-Po battery. For example, 3 cells Li-Po battery have 11.1 Volt. This is because each cell contains approximately 3.7 Volt. In general, the exact voltage should be written on the battery pack.

The C rate: C represents the capacity of a battery divided by 1 hour, its units are amps. It represents a 1 hour discharge rate using the nominal capacity of the battery. So a discharge rate of 10C for a 5Ah battery would be 50 amps. The concept of 'C' is also used for charge currents, since both charge and discharge properties are proportional to the capacity of the battery, so a 5C charge rate for a 5 Ah battery would be 25 amps. Most portable batteries are rated at 1C. This means that a 1000mAh battery would provide 1000mA or 1 amps for one hour if discharged at 1C rate. The same battery discharged at 0.5C would provide 500mA for two hours. At 2C, the 1000mAh battery would deliver 2000mA for 30 minutes. 1C is often referred to as an one-hour discharge; a 0.5C would be a two-hour, and a 0.1C a 10-hour discharge. Figure 4.8 shows the different cell voltage level and discharge time when uses different C-rate.

Capacity of Battery – mAh: The unit uses in figuring the battery capacity is Amp-hour (Ah), which is defined as the amount of current that a battery can deliver for 1 hour before the battery voltage reaches the end-of-life point. In batteries for mobile device, milliamps hour (mAh) is commonly used to describe the total amount of current a battery can store. A higher mAh rating means the battery can power a device that consumes more power and/or for a longer amount of time. Since a battery changes voltage during the discharge, it is not a perfect measure of how much energy is stored, for this engineers use watt-hour (WHr) instead. So Amp-Hours, (Ah), or milliamp-Hours (mAh) is a measure of the size of the battery a 10 mAh battery has half the capacity of a 20 mAh battery, even though they may be in the same physical package.

Capacity of Battery – Percentage of Capacity (capacity%): The capacity of a battery is commonly measured with a battery analyzer. If the analyzer's capacity readout is displayed in percentage of the nominal rating, 100% is shown if a 1000mAh battery can provide this current for one hour. If the battery only lasts for 30 minutes before cut-off, 50% is indicated. A new battery sometimes provides more than 100% capacity.

Nominal Capacity: The nominal voltage of a galvanic cell is fixed by the electrochemical characteristics of the active chemicals used in the cell, the so called 'cell chemistry'. The actual voltage appearing at the terminals at any particular time, as with any cell, depends on the load current and the internal impedance of the cell and this varies with, temperature, the state of charge and with the age of the cell.

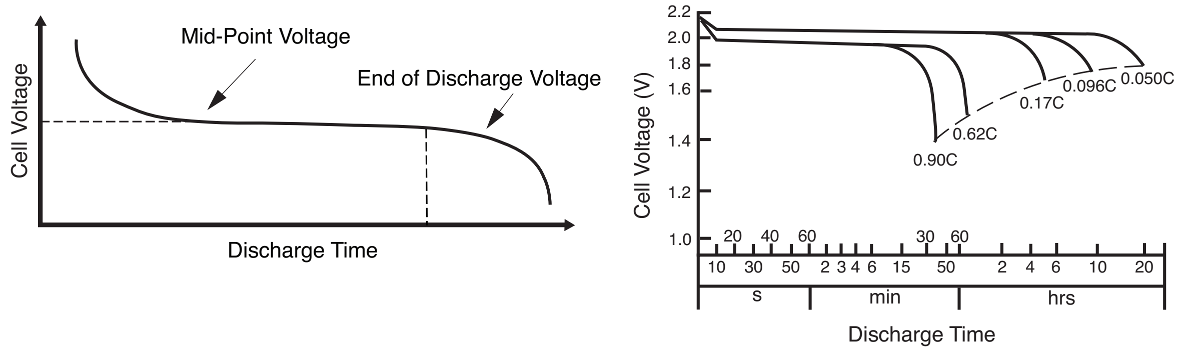
The mid-point voltage (MPV): is the nominal voltage of the cell, and is the voltage that is measured when the battery has discharged 50% of its total energy. The measured cell voltage at the end of its operating life is called the EODV, which stands for End of Discharge Voltage (some manufacturers refer to this as EOL or End of Life voltage). Both MPV and EODV are shown in Figure 4.8.

The gravimetric energy density of a battery: is a measure of how much energy a battery contains in comparison to its weight.

The volumetric energy density of a battery: is a measure of how much energy a battery contains in comparison to its volume.

Discharge: Discharge is the conversion of the chemical energy of a cell into electrical energy, which can then be used to supply power to a system.

Discharge curve: Discharge curve is a plot of cell voltage over time into the discharge, at a constant temperature and constant current discharge rate (see Figure 4.8).



(a) Discharge Curve of different cell batteries

(b) Discharge Curve at different C-rate

Figure 4.8: General Discharge Curve Characteristic

Energy (Watt-hours): While mAh measure of how much current a battery can provide over time, the more practitioners concern about is how much total energy a battery can deliver. Energy is measured in Watt-hours (Wh), the product of voltage and current over time, or volts times amperes, measured over hours. Equation 4.3 below shows the relationship between Watts, Volts and Amps.

$$Watts(Wh) = Volts(V) \times Amps(Ah) \tag{4.3}$$

To measure total energy, we need to measure the voltage and current moment by moment throughout the battery's discharge, multiply the two values together, and total up all the individual readings.

Rechargeable batteries used for mobile devices The purpose of a battery is to store energy and release it at the appropriate time in a controlled manner. There are two categories of battery cells, Primary Cells and Secondary Cells. Rechargeable battery is the secondary cell while primary cells are not rechargeable (disposable). There are many distinct of battery chemistry in the market used for secondary cells, such as NiCad, NiMH, Lead Acid, Li-ion, Li-polymers etc. Table 4.1 shows the basic information on the battery made of different chemistry substances mentioned above.

Table 4.1: General specification of different battery chemistry type

	NiCad	NiMH	Li-ion	Li-Polymers
Gravimetric Energy Density (Wh/kg)	45-80	60-120	110-160	100-150
Cell Voltage (Nominal Average)	1.20V	1.25V	3.7V	3.7V
Load Current* - peak - best results	 20C 1C	 5C 0.5C or less	 >20C 5C or less	 >20C 5C or less
Size (physically	AA (0.57x1.99 inch)	AA (0.57x1.99 inch)	Dimension Variable	Dimension Variable
* Multiply the mAh rating by the "C" value to arrive at the peak current. For example a 1000 mAh battery with a 20C rating can supply a peak current of 20,000 mA = 20 A.				

Each cell chemistry has its own characteristic nominal voltage and discharge curve. Some substances such as Lithium-ion have a fairly flat discharge curve while others such as Lead acid have a pronounced slope. The power delivered by cells with a sloping discharge curve falls progressively throughout the discharge cycle. A flat discharge curve presents the supply voltage that stays reasonably constant throughout the discharge cycle. The Figure 4.9 shows the different discharge curve of different cell chemistry.

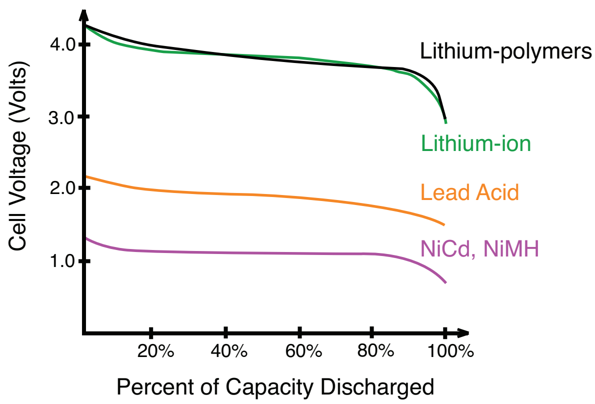


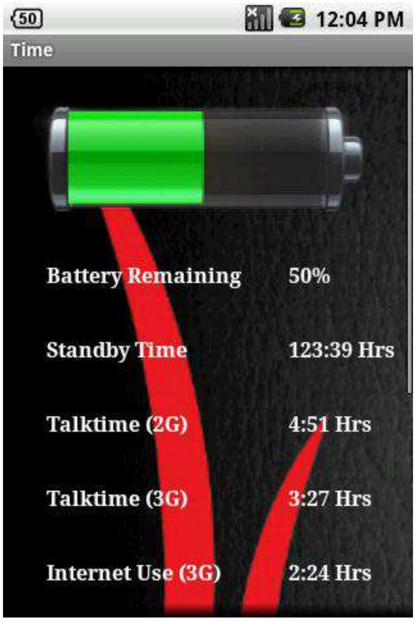
Figure 4.9: discharge characteristic of different cell chemistry [2],[3]

From Figure 4.9, Lithium-ion has a far higher voltage than the others. This implies that Lithium-ion should deliver more energy or longer in operation hours under the same capacity (mAh) of battery according to Equation 4.3. However, Figure 4.9 comparing those different cell chemistry in capacity discharged percentage cannot reflect how long the battery will last. Actually, the discharge curve gives us the idea of capacity (mAh) and voltage (depending on cell chemistry substance) when a battery is discharged at a specific and constant discharged rate. To the context of our study, mobile nodes have variety of activities which require vary level of current (non-constant discharging characteristic). Most of the mobile nodes in today's market always advertise their product with the battery characteristic. One of the most popular characteristics is the time-usage over a different underlying technologies and activities. The example of some mobile nodes' batteries specification can be found inside Table 4.2.

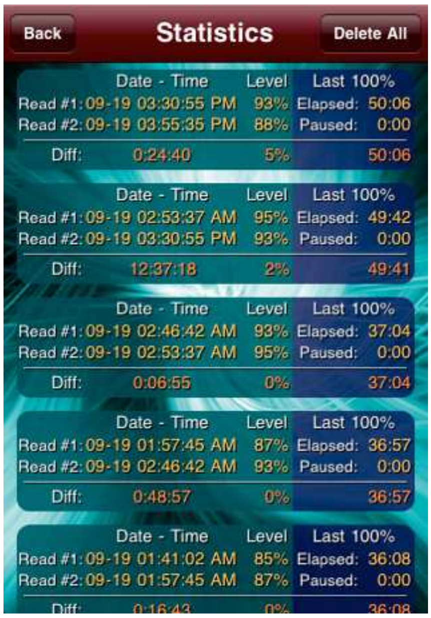
Table 4.2: Example of specification of mobile nodes' batteries

Mobile Node Models	Battery features			
	Type	Watt-Hour (WHr)	Time Usage	Capacity (mAh)
iPhone 3Gs	Lithium-ion	no report	14 hrs / Talk-time / 2G 7 hrs / Talk-time / 3G 300 hours / standby time	1219 [121]
HTC desire	Lithium-ion	no report	400 mins / Talk-time / 2G 390 mins / Talk-time / 3G 340 hrs / standby time	1400
BLS-4 (Nokia Cell Phone)[122]	Lithium-ion	no report	8 hrs / Talk-time 100 hrs / standby time	1500
Samsung OMNIA II	Lithium-ion	no report	12 hrs / Talk-time / 2G 8 hrs / Talk-time / 3G 430 hrs / standby time	1500
Latitude D620 [123]	Lithium-polymers 6-cell	48	no report	no report
MacBook Pro [124]	Lithium-polymers	77.5	8-9 hrs wireless productivity	no report

Most of current mobile nodes use Lithium-ion or Lithium-polymers which have a very similar characteristics (see Figure 4.9 and Table 4.1). The aim of this study is to consider the available battery capacity as a criteria for finding a robust topology in DT-MANETs. Since, most of mobile nodes use the similar batteries (Li-ion and Li-Po), the most interesting characteristic is the remaining capacity or available energy of the battery in mobile node. To date, many applications found in mobile nodes (i.e., iPhone, Android phone, etc.) are for monitoring battery and its statistic information such as the remaining Talk-time (via using 2G, 3G, and wifi) and standby time. Figure 4.10(a) and 4.10(b) depict examples of such applications.



(a) Battery status provided by an android application



(b) Battery statistics provided by an iPhone application

Figure 4.10: Screenshot examples of battery applications in mobile phones

The robust topology in a DT-MANET should equip with enough energy nodes to provide effort in forwarding messages on behalf of other nodes. Nodes with higher battery level are more likely to give longer contribution to network. In other words, the lifespan of network strongly depends on the cooperation between mobile nodes, and the higher battery nodes are more reliable to operate longer than lower battery nodes. The low energy nodes should not act as a forwarder (disseminator) since the communication is likely to fail. Instead, having a high battery node as forwarder is more secure.

The percentage characteristic has been selected and used as the quality of node due to two main reasons. First, the variety battery types and capacities currently found in market. As summarized in Table 4.2, nodes may have different in types of cell which leads to the different capabilities (found in Table 4.1). Second, In general, the more functionality nodes such as laptop or tablets have a higher capacity battery due to the high consumption according to the functionality. In fact, this higher capacity of battery in high functionality nodes does not imply the longer contribution to network. Thus, percentage of batter level left in each node is used. Furthermore, this particular value can be observed easily in different node types via some applications (examples are shown in Figure 4.10).

4.3.3 Capacity Bandwidth Aspect

Nodes in an ad hoc wireless network share a common broadcast radio channel. Since the radio spectrum is limited, the bandwidth available for communication in such networks is also restrained. Bandwidth represents the capacity of the connection. The greater the capacity, the more likely that greater performance will follow, though overall performance also depends on other factors, such as latency. The capacity bandwidth of each communication edge gives us an immediate idea of how well the edge prompts to transfer data. From an end-to-end communication perspective, the data cannot travel any faster than the smallest bandwidth link involved between two entities. In this work, such the smallest bandwidth link is also called the bottleneck bandwidth edge. Similar to the trust value, the bandwidth capacity of each edge is assumed and has the same specific range as the trust value range. The higher the capacity bandwidth, the more data can be transfer through this edge.

Definition 45 (Capacity Bandwidth).
In a communication graph $G = (V, E)$, we denote $CB(v_1, v_2)$ a capacity bandwidth value of edge $(v_1, v_2) \in E$.

In graph theory, an end-to-end communication is represented by the notion of Path. A path in a graph is a walk in which all edges are distinct. A walk in a graph is an alternating sequence of vertices and edges, beginning and ending with a vertex, in which each vertex (except the last) is incident with the edge which follows it and the last vertex is incident with the preceding edge. It is *closed* if and only if the first vertex is the same as the last and otherwise *open*. For any tree and acyclic graph, it is obvious to have only open path, where the first and the last node are differ. The formal definition of Path is given below.

Definition 46 (Path).
Given $P(a, b)$ a path from a vertex a to a vertex b in a graph $G = (V, E)$. A path, $P(a, b)$, is a sequence $\{v_0, v_1, v_2, \ldots, v_k\}$ of vertices such that $a = v_0$, $b = v_k$, and $(v_{i-1}, v_i) \in E$ for $i = 1, 2, \ldots, k$. The path contains vertices $v_0, v_1, \ldots, v_k$ and the edges $(v_0, v_1), (v_1, v_2), \ldots, (v_{k-1}, v_k)$. The length of the path is the number of edges in the path. There is always a 0-length path from u to u.

4.3.4 Discussion on other criteria

4.3.4.1 Signal Strength

In telecommunications, particularly in radio, signal strength refers to the magnitude of the electric field at a reference point that is a significant distance from the transmitting antenna. It may also be referred to as received signal level or field strength. For this study, signal strength refers to the received signal level at a node A from a node B. For very low-power systems, such as mobile phones, signal strength is usually expressed in dB-microvolts per meter (dBμV/m) or in decibels above a reference level of one milliwatt (dBm).

Signal strength is one of an interesting criteria to be used for selecting robust tree topology. Dube et al. [125] proposed the signal stability based adaptive routing for MANETs. This work uses the signal strength criterion to differentiate weak and strong signal strength. Together with signal stability criterion, [125] selects the edge that has a strong signal strength and have been in existence for a time greater than some threshold. As can be seen in the previous work, signal strength is not suitable to be used as a criterion alone. Instead, signal strength has been used together with signal stability.

In general, signal strength at a specific point can be determined from the power delivered to the transmitting antenna, its geometry, and radiation resistance. Weak reception signal strength can also be weakened by noise. One main drawback of utilizing signal strength for this study is a single point of failure. Given that one node has a very strong signal strength (due to very high power to transmitting antenna and less noise), it is more likely that this node can cover farther area in the network. As a result, multiple nodes may be under this node coverage range. Due to its strong signal, these neighbors willingly connect to this particular node to form a high signal strength spanning forest. However, the higher the signal strength also means the more usage of battery power. This high signal strength node may have short of power very soon. The disappearance of such node, where many nodes connect to it as a hub, severely destroys the tree into pieces. This is the reason we are not interested in this criterion for this study.

4.3.4.2 Age of edges

Age of edges measures the stability level of each edge after its appearance into the network. This criterion has been used mainly in dynamic network study [91]. This study also use this metric for studying the characteristic of network used for experimentation. However, it has been used in an offline mode. The computation method of this metric is not suitable to apply in an online mode (each node hold the computation along the simulation time) because a large number of memories must be used to track all edge of a node. Since this study aims at managing topology in DT-MANETs, memory space in each mobile node is assumed to be small. Thus, the usage of metric ‘age of edges’ is not suitable.

Summary In this section, multiple quality criteria have been discussed. Age of edges metric might be useful but are not suitable for working in DT-MANETs where memory and computation are limited. Signal strength metric is not suitable to be used since it might expose the one point of failure problem. In summary, three robustness quality criteria are selected. They are trust level, battery level and capacity bandwidth.

4.4 Robustness Metrics

As discussed under previous section, nodes with higher quality level are more likely to be able to complete their tasks than lower ones. Similarly, edges with higher quality tends to give a higher quality of service than lower ones. Thus, the main idea of robust spanning tree is to have high quality nodes acting as router (forwarder) and less quality nodes being at leaves position of tree. On the other hand, less quality edges should not be included in the tree structure if not necessary.

In order to determine robust spanning trees when concerning nodes' quality in particular, [SMCia08] introduces quality measurement by means of two different metrics. These are $nodeWeight()$ and $isolateLowQualityNode()$. Another robustness metric namely as $pathWeight()$ function is introduced to measure the quality of robust spanning trees when concerning edges' quality. We regard these three different metrics as robustness metrics from this point onward. These metrics are described with examples under this section.

In order to summarize the quality of the created spanning tree, the value of functions from different studied algorithms will be compared where a higher value indicates a superior quality.

4.4.1 nodeWeight() function

Nodes with higher quality level are more likely to be able to complete their tasks than lower ones. The $nodeWeight()$ function is introduced in [SMCia08] to assess robust spanning trees with respect to this objective. Having $V(\gamma)$ as the set of all nodes in a tree γ, the $nodeWeight()$ function of a trusted spanning tree can be determined by the following equation:

$$nodeWeight(\gamma) = \sum_{x \in V(\gamma)} quality(x) \times tree\_degree(x) \tag{4.4}$$

The function $quality(x)$ represents a particular quality that the robustness concerning about at any moment, while $tree\_degree(x)$ represents the number of one-hop neighbors of node x in the tree (emphasize here that concerning tree neighbors and not physical neighbors). Let use the trust value of node x ($trust(x)$) as $quality(x)$ as example.

Figures 4.11(a) and (b) are examples to illustrate how the $nodeWeight()$ function can assess this quality where the $threshold$ used in this particular example equals to 0.2. In Figure 4.11(a), the node with lowest trust level gets the highest $tree\_degree$, while the node with highest level gets the lowest $tree\_degree$ (i.e the node A has a trust level of 0.2 and $tree\_degree$ of 3, while the node E has a trust level of 1 and $tree\_degree$ of 1), hence the $nodeWeight(\gamma_a)$ function for this trusted spanning tree is 4.4. Figure 4.11(b) depicts the opposite (i.e. the node with the highest trust level possesses the highest $tree\_degree$ (node E), while the node with the lowest level possesses the lowest $tree\_degree$ (node A)). The $nodeWeight(\gamma_b)$ function for this trusted spanning tree is 6.8. In order to measure this function on the whole graph (the spanning forest), equation 4.5 is introduced below.

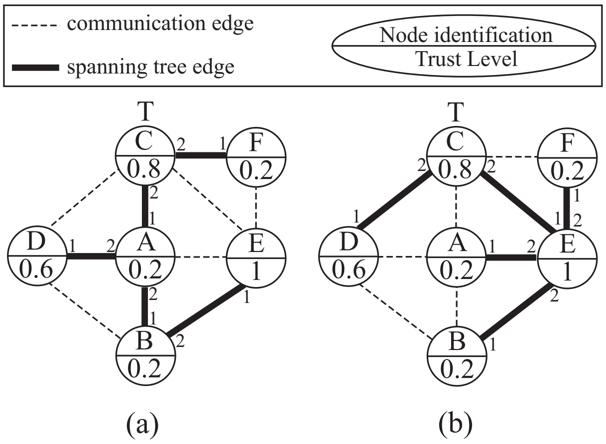


Figure 4.11: An example scenario for Illustrating the details of *nodeWeight()* and *isolateLowQualityNode()* cost functions

$$nodeWeight(G(t)) = \sum_{\gamma \in G(t)} nodeWeight(\gamma) \tag{4.5}$$

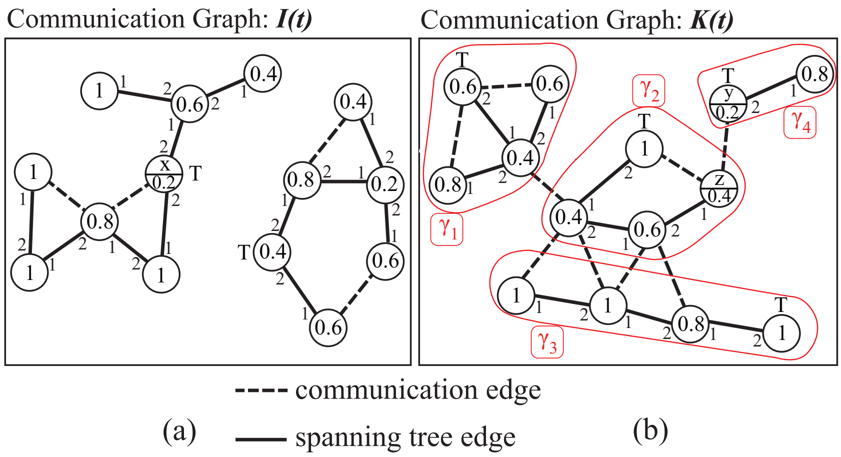


Figure 4.12: An example scenario for Illustrating the details of both efficiency and robustness metrics

Based on the Figure 4.12, $nodeWeight(I(t))$ gives 14.8, while $nodeWeight(K(t))$ gives 13.2.

4.4.2 isolateLowQualityNode() function

Since low trust level nodes are unreliable and have tendency to break away from the network, allowing them to have high degrees in the tree will degrade the robustness of the tree and increase the likelihood of disconnection from the spanning tree. Therefore, in order to maximize the quality of service and minimize the re-connecting task, nodes with lowest trust levels should be assigned the lowest *tree_degree* position in the tree. $isolateLowQualityNode()$ is introduced as a mean to assess spanning forest with respect to this objective.

Example using trust level of cooperative enforcement paradigm, this function evaluates the efficiency of a trusted spanning forest by noting how well it can isolate non-trustable nodes n' where $n' \in V(G(t))$ and $trust(n') \leq threshold$. The function measures the percentile of n' nodes at leaf. The higher value of $isolateLowQualityNode()$ function signifies better quality trusted spanning tree. Let $\Theta^*(\gamma) = \{n' \in \Theta(\gamma) \mid tree\_degree(n') = 1\}$ be the set of low trustable nodes being leaves in the tree γ. The $isolateLowQualityNode()$ function is defined by:

$$isolateLowQualityNode(\gamma) = \left(\frac{\mid \Theta^*(\gamma) \mid}{\mid \Theta(\gamma) \mid} \right) \times 100 \tag{4.6}$$

Hence, the *isolateLowQualityNode()* value for Figure 4.11(a) is 33.33% while this value is 100% for Figure 4.11(b). Equation 4.7 is introduced below as a mean to measure the same objective in a spanning forest. The result given by this equation for Figure 4.12(a) and (b) equal to 0% and 100% respectively.

$$isolateLowQualityNode(G(t)) = \left(\frac{\sum_{\gamma \in G(t)} \mid \Theta^*(\gamma) \mid}{\sum_{\gamma \in G(t)} \mid \Theta(\gamma) \mid} \right) \times 100 \tag{4.7}$$

The more detailed explanation is as follows. Both graphs $I(t)$ and $K(t)$ have the same number of nodes and the same proportion of trust values of nodes, but different forest topologies. Within each graph, there exists one low trusted node according to the given threshold value. The location of this low trusted node (y) in $K(t)$ is good at this time t since it is at leaf of tree γ_4, while the low trusted node (x) in $I(t)$ is in middle of a tree, having two trusted spanning tree edges.

However, the *spanningRatio()*, one of the efficient metrics, of $K(t)$ is 0.25, which means the current good value of $isolateLowQualityNode()$ is not valid to the objective. If there are only few changes in the forest topology and node y and z of $K(t)$ can merge in several time steps later, $isolateLowQualityNode(K(t))$ will become 0%. This case also shows the impact of efficiency metric like *spanningRatio*() has on other metrics.

4.4.3 pathWeight() function

Has its focus on an end-to-end communication, the quality measurement introduces by $pathWeight()$ function measuring all possible paths in a constructed spanning tree. Since the low value of one edge can effect the quality of all the whole communication path as it is the bottle neck edge, pathWeight() function interests only the minimum value (the bottleneck value) of each path. The higher the sum value of all path, thus means the better the quality of the tree and the more robust the tree structure.

Each edge is given a quality. We denote $MinQuality(P(a,b))$ as the minimum quality value of any edge belonging to path $P(a,b)$. The robustness metric pathWeight() function is defined as follow:

$$pathWeight(\gamma) = \sum_{\forall(a,b)\in E(\gamma)} MinQuality(P(a,b)) \tag{4.8}$$

In order to measure this function on the whole graph (the spanning forest), equation 4.9 is introduced below.

$$pathWeight(G(t)) = \sum_{\gamma\in G(t)} pathWeight(\gamma) \tag{4.9}$$

Thus, the robustness metrics are reviewed here briefly using example Figure 4.13. In the figures, value of quality is presented in the middle of each edge. The spanning tree in Sub-Figure 4.13(a) (γ_a) comprises of five edges where the minimum quality among these five edges is 0.8. This is the least quality value. Further investigating on this sub-figure, we found that the communication from node F using spanning tree to any other nodes in this connected subgraph will utilize maximum quality at 0.8. Meanwhile, the communication occurs between node A, C and D can use the maximum quality of edge at 1. Actually, the spanning tree 4.13(a) gives us the best spanning tree regarding to the given quality criteria. Since the minimum quality on this tree is 0.8, it is guaranteed that the communication can use the quality of edge at least 0.8 of quality level.

On the contrary, Figure 4.13(b) gives us a spanning tree with the minimum quality at 0.4. This means many communications are limited their quality usage to this value. The pathWeight() function calculates all possible communication paths within a single tree. The results of the metric are 25.6 and 16.4 for figure 4.13(a) and (b) respectively.

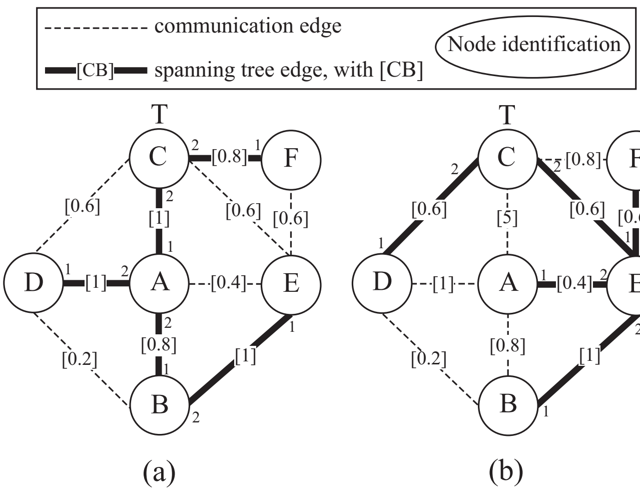


Figure 4.13: An example scenario for Illustrating the details of robustness metric of pathWeight() function

4.5 Robust Spanning Forest by Nodes and Edges' qualities

4.5.1 Problem Description: Cooperative Enforcement Aspect

According to the cooperative enforcement paradigm, nodes with higher trust level are more likely to be able to complete their tasks than lower ones (i.e., by asking a high trust level node to forward a message to destination D, it is likely that the message will arrive at D). This means having high trusted nodes as forwarders, the communication is likely to be succeeded. On the other hand, low trusted nodes are unreliable. The low trusted node may be often switched on and off due to some physical problems or may skip the forwarding task intentionally. Thus, having nodes with low trust level localized on leaves is advantageous since they would not be responsible for forwarding information to others. Hence, the unsuccessful communication will be reduced. Moreover, losing them at these positions has little effect on the overall structure. Thus, robust trusted spanning tree is a tree which has low trusted node(s) as close as to leaf of tree.

Trust level of a node n, denoted by $trust(n)$, where $0 < trust(n) \leq 1$, defines the levels of quality of services it can provide. Whether a node n can be trusted is determined by a given threshold. Let $\Theta_t = \{n' \in V_t(G) | trust(n') \leq threshold\}$ be the set of all low-trustable nodes at moment t.

Each node is given a quality using trust level ($trust()$). We are looking for a set of spanning trees, (Γ^*), in a communication graph at moment t, $G(t)$ such that:

$$
\begin{aligned}
nodeWeight(\Gamma^*) &= nodeWeight(G^*) \\
&= \max_{\forall \gamma \in G(t)} \left(\sum_{x \in V(\gamma)} quality(x) \times tree\_degree(x) \right)
\end{aligned}
$$

where $tree\_degree(x)$ represents the number of one-hop neighbors of node x in the tree.

and

$$
\begin{aligned}
isolateLowQualityNode(\Gamma^*) &= isolateLowQualityNode(G^*(t)) \\
&= \max_{\forall \gamma \in G(t)} \left(\frac{\sum_{\gamma \in G(t)} \mid \Theta^*(\gamma) \mid}{\sum_{\gamma \in G(t)} \mid \Theta(\gamma) \mid} \right) \times 100
\end{aligned}
$$

where $\Theta^*(\gamma) = \{n' \in \Theta(\gamma) \mid tree\_degree(n') = 1\}$ be the set of low trustable nodes being leaves in the tree γ.

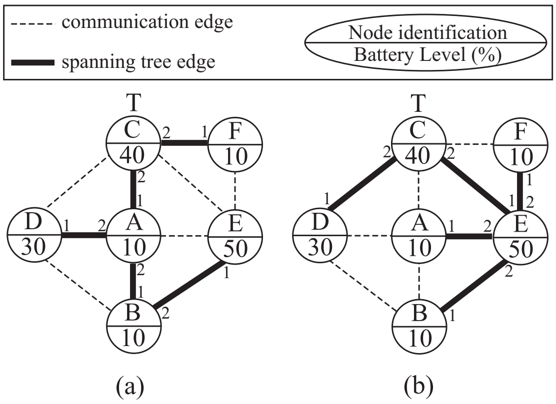


Figure 4.14: An example scenario for Illustrating the details of robustness metrics for percentage of battery level case

4.5.2 Problem Description: Energy Availability Aspect

The robustness metrics are reviewed here briefly using example Figure 4.14. Nodes with higher batter level left are more likely to continue functioning inside network. Thus, having such high battery node acts as forwarder can help increasing robustness to the tree. On the other hand, low battery nodes tend to shut down soon and then break away from the network. Encouraging low battery nodes to situate at leaf of tree helps reducing re-connecting tree. The concerning quality of node n, $quality(n)$, is the percentage of battery level on node, where the range of value is $0 \leq battery(n) \leq 100$. The threshold used to illustrate this example figure is at 20%. Thus, let $\Theta_t = \{n' \in V_t(G) | battery(n') \leq threshold\}$ be the set of all low-battery nodes at moment t. The nodeWeight() function (4.4) of γ_a and γ_b equal to 220 and 350 respectively. However, both Figures 4.14 give the same spanningRatio() at 100% since each graph comprises of one connected component and the tree spanning cover all node. For isolateLowQualityNode() function, Figure 4.14(a) give 33% while Figure 4.14 is perfectly positioning both low and high battery level nodes and give 100%.

Thus, the formal problem description is defined as follow. Each node is given a quality using percentage of battery level. We are looking for a set of spanning trees, (Γ^*), in a communication graph at moment t, $G(t)$ such that:

$$
\begin{aligned}
nodeWeight(\Gamma^*) &= nodeWeight(G^*) \\
&= \max_{\forall \gamma \in G(t)} \left(\sum_{x \in V(\gamma)} quality(x) \times tree\_degree(x) \right)
\end{aligned}
$$

where $tree\_degree(x)$ represents the number of one hop neighbors of node x in the tree.

and

$$
\begin{aligned}
isolateLowQualityNode(\Gamma^*) &= isolateLowQualityNode(G^*(t)) \\
&= \max_{\forall \gamma \in G(t)} \left(\frac{\sum_{\gamma \in G(t)} \mid \Theta^*(\gamma) \mid}{\sum_{\gamma \in G(t)} \mid \Theta(\gamma) \mid} \right) \times 100
\end{aligned}
$$

where $\Theta^*(\gamma) = \{n' \in \Theta(\gamma) \mid tree\_degree(n') = 1\}$
be the set of low trustable nodes being leaves in the tree γ.

4.5.3 Problem Description: Capacity Bandwidth Aspect

The aim of this study is to construct robust topology of spanning forest by focusing on the capacity bandwidth (CB) of selected edges. Since the higher the capacity bandwidth regulate the higher quality of service, the spanning forest with a set of high CB edges are preferred. However, the end-to-end communication or communication path are limited its capacity by the edge with smallest CB value. Thus, this particular problem is to find a spanning tree of a connected component which maximize the minimum CB of all possible path.

Each edge is given a weight using a capacity bandwidth (CB). We denote $MinCB(P(a,b))$ as the minimum value of the CB of any edge belonging to $P(a,b)$. We are looking for one Path, $P^*(a,b)$, such that:

$$
MinCB(P^*(a,b)) = \max_{\forall (a,b) \in V(\gamma)} (MinCB(P(a,b))) \tag{4.10}
$$

4.6 Solutions for Robust Spanning Forest in DT-MANETs

4.6.1 Introduction

Providing robust spanning forest is subjective to the problem description of being robust. As discussed earlier, robust topology management concerns multiple criteria and several level of robustness. Since different desire of being robust may differ from one situation to another, this study provide solutions in terms of framework where each module can be seen as plugin. The framework comprises of (1) a set of robust criteria for topology management problem, (2) the evaluation methodologies for different solutions of different problems (e.g., single-objective and multi-objective problem). and (3) a set of distributed algorithms as solutions providing robust spanning forest in DT-MANETs. So far the robustness criteria on both node and edge has been discussed, as well as, the efficiency and robustness metrics. Figure 4.15 illustrates the framework at the present stage. It shows only what has been presented, while the non-presented module are under gray color.

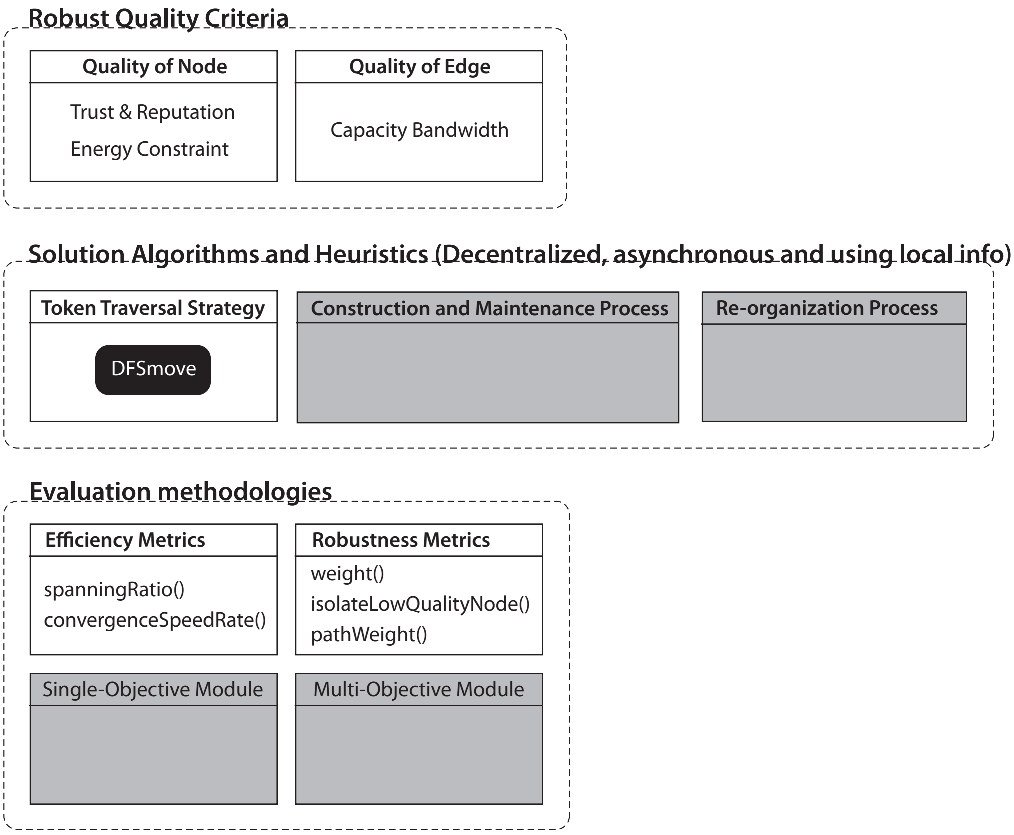


Figure 4.15: Framework proposed by this study including Robust Quality Criteria, Token Traversal Strategy and Metrics for Efficiency and Robustness

The construction of spanning tree using token management concerns mainly two processes; (1) Token Traversal Strategy and (2) Merging Process (as discussed under section 4.2.3. This study proposed DFSmove as a better solution for token traversal strategy discussed previously under section 4.2.4.3. Under this section, solution algorithms and heuristics for construction and maintenance process are explained. These construction and maintenance algorithms aim at providing a higher robustness of spanning tree. The main activity that these algorithms focus on is the merging process. However, after trees merged the dynamic of both topological structure of network and the quality of nodes and edges may change. The re-organization process is needed in such dynamic environment so that the topology can adapt itself to the continuous changes. The re-organization algorithms are also discussed in this section as well.

4.6.2 Construction and Maintenance Process

4.6.2.1 G-NODE

'Greedy Robust Spanning Tree' or G-NODE [EUC08] is a decentralized algorithm that employs greedy algorithm for constructing spanning trees in a dynamic network like DT-MANETs. This algorithm attempts to create robust spanning forest by relying on quality level of each neighbor.

As higher quality nodes are more likely to provide higher quality of service, therefore designating them as interior/inner nodes (where they can be assigned to relay messages to other nodes when needed) is preferable. Excepting the fact that G-NODE utilizes quality level of nodes, it is an extension of the original algorithm (DA-GRS algorithm). In G-NODE the rendez-vous assumption of the original algorithm is relaxed. Thus, in G-NODE, several tokens can meet simultaneously. Furthermore, by relying on 'Greedy' heuristics, G-NODE can select the highest quality node to merge when applicable. The merging operation in G-NODE is described in algorithm 4.

Algorithm 4 **Look for other tokens around token** τ_i

```
1: τ^best is token owned by the most quality neighborhood
2: if τ^best ≠ ∅ then
3:     Merge_With(τ_i, τ^best) //merge the two tokens
4: else
5:     Move_Token(τ_i) //continue to move the token randomly
6: end if
```

Indeed, the advantage of G-NODE occurs when several tokens meet. It allows each node owning a token to choose the most quality neighbor to merge. Table 4.3 summarizes the differences between DA-GRS algorithm and G-NODE.

G-NODE concerns the quality level of nodes. Thus, all robust spanning forest problems that concerns only in the quality of node can use G-NODE. In this study, both problems of cooperative enforcement and energy availability aspect use G-NODE to provide a robust spanning forest. The experimentation and results are shown in Chapter 5.

4.6.2.2 G-PATH

The heuristic called G-PATH is proposed by followed the restriction of algorithms working for topology management in DT-MANETs that we discussed earlier under section 3.3.3.3. We

Table 4.3: Summary heuristic used by DA-GRS and G-NODE at each activity concerning spanning forest construction and maintenance using token management strategy

	DA-GRS algorithm	G-NODE
Initial State	-	-
Act on Physical Disruption(1)	-	-
Act on Physical Disruption(2)	-	-
Merging Process	rendez-vous synchronisation Randomness	relax rendez-vous Greedy Heuristic
Token Traversal	Randomness	DFSmove

summarized them here again: the algorithms used for managing topology in DT-MANETs should be a distributed and localized algorithm (utilizing only local information), and communicate asynchronously. G-PATH follows the main idea of G-NODE, however it concerns the quality of edges. In G-NODE, when several tokens meet simultaneously, node has a chance to select the highest quality node (according to the specified criteria). Likewise, G-PATH relax the synchronization method of the previous algorithm of DA-GRS and given the communication protocol as realizing in section 4.2.3. G-PATH relies on greedy heuristic to select the highest quality edge to merge when applicable. The merging operation in G-PATH is described in algorithm 5.

Algorithm 5 **Look for other tokens around token** τ_i

```
1: τ^best is token owned by the neighbor who possesses the highest quality edge
2: if τ^best ≠ ∅ then
3:     Merge_With(τ_i, τ^best) //merge the two tokens
4: else
5:     Move_Token(τ_i) //continue to move the token randomly
6: end if
```

4.6.3 Re-organization Process

So far, the proposed algorithms (G-NODE and G-PATH with DFSmove heuristic) aim at functioning spanning forest in purely distributed manner for managing spanning forest topology in DT-MANETs. Indeed, the proposed heuristic and algorithms does both construction and maintenance of robust spanning forest. However, the benefit of such greedy heuristics give high efficient on the very beginning period of construction and on the severely changes occur to underlying network (i.e., most of nodes go offline or individual node moves in scatter way swiftly). The maintenance of robust topology needs to be handle carefully so that the topology continue efficient and robust.

In this section, we propose two different heuristics to adapt the existing managed topology according to the continuous changes of situation, both in terms of changes from underlying network and the changes from quality of entity (such as the changes of trust level or battery level of nodes and the capacity bandwidth changed for each edges). The maintenance heuristics or algorithms still need to work in distributed manner but yet effective at global scheme. Our proposed heuristic need only one-hop information as usual.

4.6.3.1 BREAK heuristic

The '*BREAK*' heuristic attempts to provide a mechanism to reorganize the established trees. The tree reorganization might be needed because of the fluctuation in both the communication topology and node or edge quality values which affects the robustness of the tree itself. Hence, BREAK is an extension of G-NODE enabling dynamic spanning forest reorganization. Nodes having low quality value or noticing low quality value of it adjacent edges will provoke the reorganization. Hence, when nodes arrive at time (T_break), they check the quality value of themselves. If the quality value of a node is equal or lower than the *threshold* then it apply 'BREAK' heuristic automatically. The low-quality node will make decision which tree edge will not be broken and then breaks all the other tree edges. This remaining edge simply is the tree edge which connect the low-quality node with the highest quality neighbor (having highest quality level among all tree neighbors). Since the breaking action is one of the basic event in G-NODE algorithm, the procedure after breaking will automatically resume thanks to the classical behavior of G-NODE.

One fact of this heuristic is it generates many trees over a connected component at the same time. The mechanism of heuristic aims at finding a better solution than the current one by breaking current tree. Consequently, many trees are generated and thus effect the efficiency of the managed topology. However, the more robust solution of topology will be met thanks to the original mechanism of G-NODE. Moreover, the fact that low quality nodes will be expelled out by this heuristics and many trees are generate at the same time makes the isolatedLowQualityNode() function benefitting the most. To summarize, the heuristic create smaller trees with low quality nodes at leave of tree. The trees are spanning harder, but the trees become more robust at the same time. Benefitting of such heuristic falls to the characteristic of network as such mobility model of network. If the velocity of network is slow to moderate, we expect that G-NODE will have high chance to span the tree. Otherwise, the high speed rate of movement fails any attempt to manage in any case.

The deterioration of the managed topology will occur repeatedly due to the mobility of DT-MANETs. BREAK heuristic will also try to cope with it repeatedly according to T_break value. Figure 4.16 illustrate the mechanism of BREAK heuristics. It contains multiple sub-figures: each sub-figure shows the situation at consecutive time t_i. Given a spanning tree at time t_0, two low quality nodes are presented in color nodes. Following the ultimate goal of this work, low quality nodes should situate at leave position. Sub-Figure 4.16(b) shows three nodes in unstable position and did trick the BREAK heuristic when it meet T_break such as at time t_1. Later, the re-state of spanning tree has been done by the classical G-NODE once again.

4.6.3.2 CHANCE heuristic

In contrast to the periodically action approach like BREAK heuristic, the proposed heuristic in this section is an opportunity based approach. Given that each node is well-aware of its current quality, topology (one-hop neighborhood), its edges' quality and its neighbor's

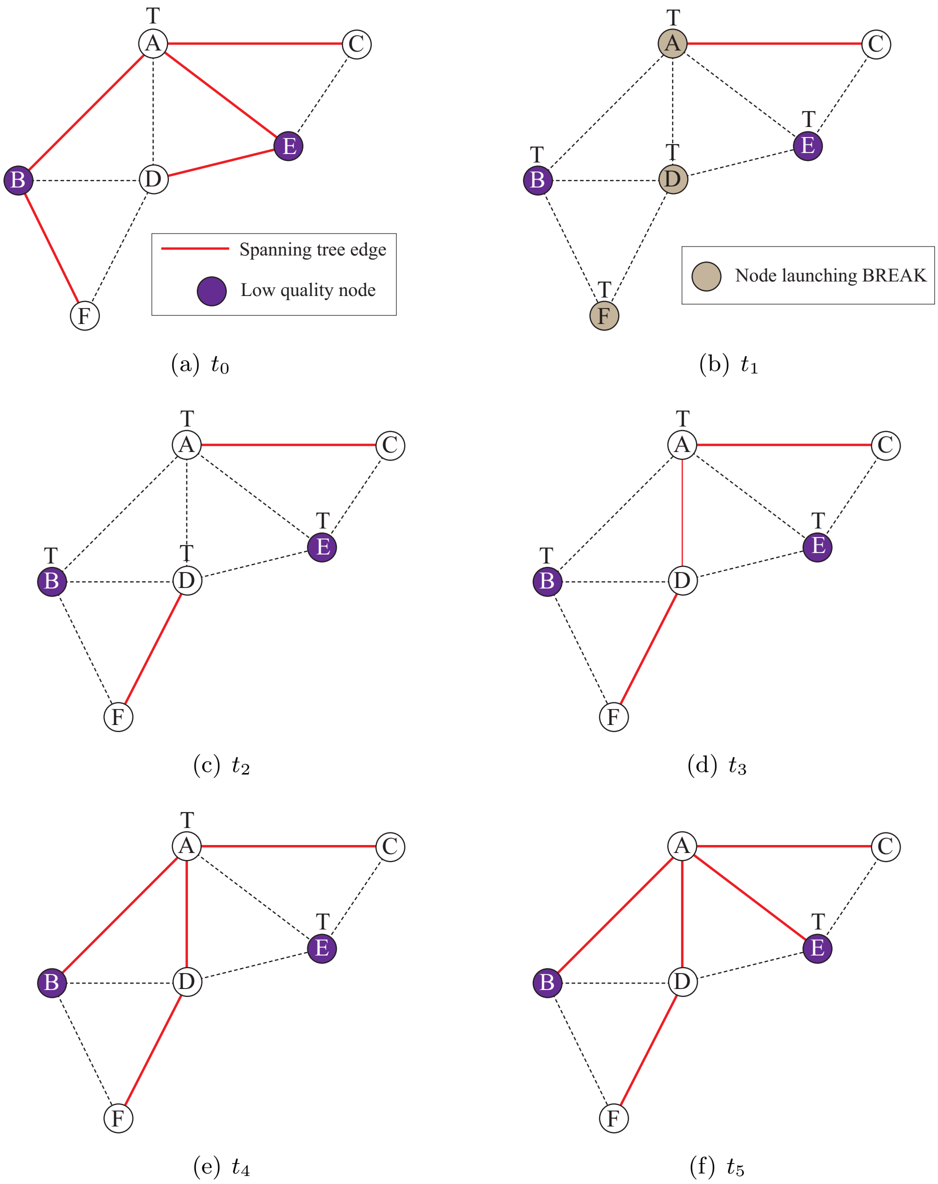


Figure 4.16: An example illustrating BREAK heuristic

quality, nodes apply CHANCE heuristic when applicable. CHANCE heuristic works for both G-NODE and G-PATH.

In case of G-NODE, nodes notice current quality summation over each edge. Given w_i quality of any node i, the quality summation over edge (v_x, v_y) is $(w_{v_x} + w_{v_y})$. We denote $QS(v_x, v_y)$ as the quality summation between node v_x and v_y. On the other hand, using CHANCE heuristic in G-PATH, we denote $QS(v_x, v_y)$ as the quality of edge (v_x, v_y). Referring to 'Merging Trees Process' described in section 4.2.3 (which described details of utilizing and applying DA-GRS reference algorithm), nodes in possession of token will announce this event by broadcasting a 'findingTk' message. Then, nodes situating within one-hop range will overhear this message regardless of being in the same tree or not. Thus, these one-hop neighbors will have knowledge of token presence nearby. At this precise moment is a chance to re-locate the created tree to become a more robust one.

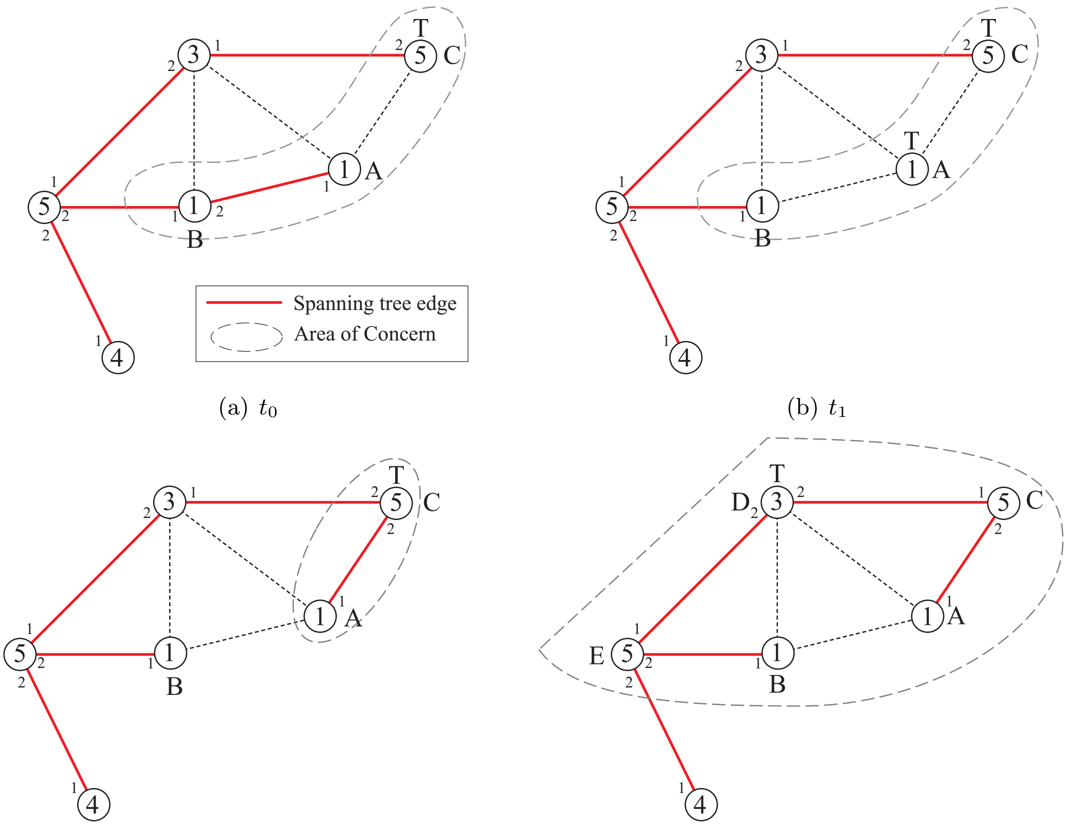


Figure 4.17: An example illustrating CHANCE heuristic in G-NODE (concerning node's quality)

Considering Sub-Figure 4.17(a), node C possesses the token and a 'findingTk' message is broadcasted. Accordingly, node A is aware of the token presence at node C, showing a chance to reform topology of the network. Node A is currently connected with node B in a tree and node B is parent of node A. The $QS(A, B)$ equals to 2. Considering merging node A with C, $QS(A, C)$ equals to 6. Since, $QS(A, C) > QS(A, B)$ and B is parent of A. Thus, A send a break message to node B and generates new token. Then, A send acknowledgement back to C regarding 'findingTk' for participating in merging process with node C. Please note that C will have priority to select a node giving highest quality according to G-NODE algorithm. Thus, it is not necessary that C must merge with A if C find another high valuable node. Since node A is child of node B and the existence of token is not in child's direction, thus node B does not regenerate token but only notice the breaking off of node A. On the other hand, node A generates new token and wait for 'SYN/ACK_finding' message to merge with C as shown in Sub-Figure 4.17(b) and 4.17(c). Since there is no other competing nodes against node A, finally, node C merges with A. Later, token moves to node D in Sub-Figure 4.17(d) where node A and B will have chance to re-consider their position. Let us consider for node A first. Node A is connecting in tree with node C which $QS(A, C)$ is 6. If A merge with D, $QS(A, D)$ equals to 4. Since $QS(A, C) < QS(A, D)$, node A will consider this is not a chance to improve its situation. Considering at position of node B, $QS(B, E)$ is 6 and $QS(B, D)$ is 4. Then, node B will not consider to take chance since $QS(B, D) < QS(B, E)$.

In summary, CHANCE heuristic is the process where each node await for its chance to improve its position in the current topology of spanning forest. The better position can be judged by comparing the current QS value ($QS_{current}()$)and the QS value of the expected position ($QS_{chance}()$). The chance is the presence of token in its one-hop neighbors. When the chance is validated to be valid by a node α such that token presence in one-hop neighbors and

$$QS_{current}(\alpha) < QS_{chance}(\alpha). \tag{4.11}$$

Then, node α send 'break' message to its parent tree neighbor (This parent tree neighbor is shown in Figure 4.17 using label '2' which is according to DA-GRS model) and send 'ACK_finding' message to token possession node. Later, node α wait for 'SYN/ACK_finding' message for confirming to merge with the chance node according to protocol defined in Section 4.2.3. The breaking process is complete when its parent send 'ACK_Break' message back to acknowledge the break action. If its parent still is in its neighborhood but did not reply with 'ACK_Break' back within a defined time, Node α will resend 'break' message to its parent again.

4.7 Multi-objectives Approach for Managing Topology in DT-MANETs

Previously in Section 3.5, we discussed roughly about robust criteria and the effect of conflicting criteria toward the importance of multi-objective optimization for managing topology in DT-MANETs. Previously in this chapter, all criteria focused in this study have been described and discussed. In summary, these criteria are trust level of nodes, battery level of nodes and capacity bandwidth of edges. The problem descriptions, evaluation metrics and algorithm solutions has been proposed and discussed earlier in this Chapter. Figure 4.18 summarizes all module we proposed into the framework so far for constructing robust spanning forest.

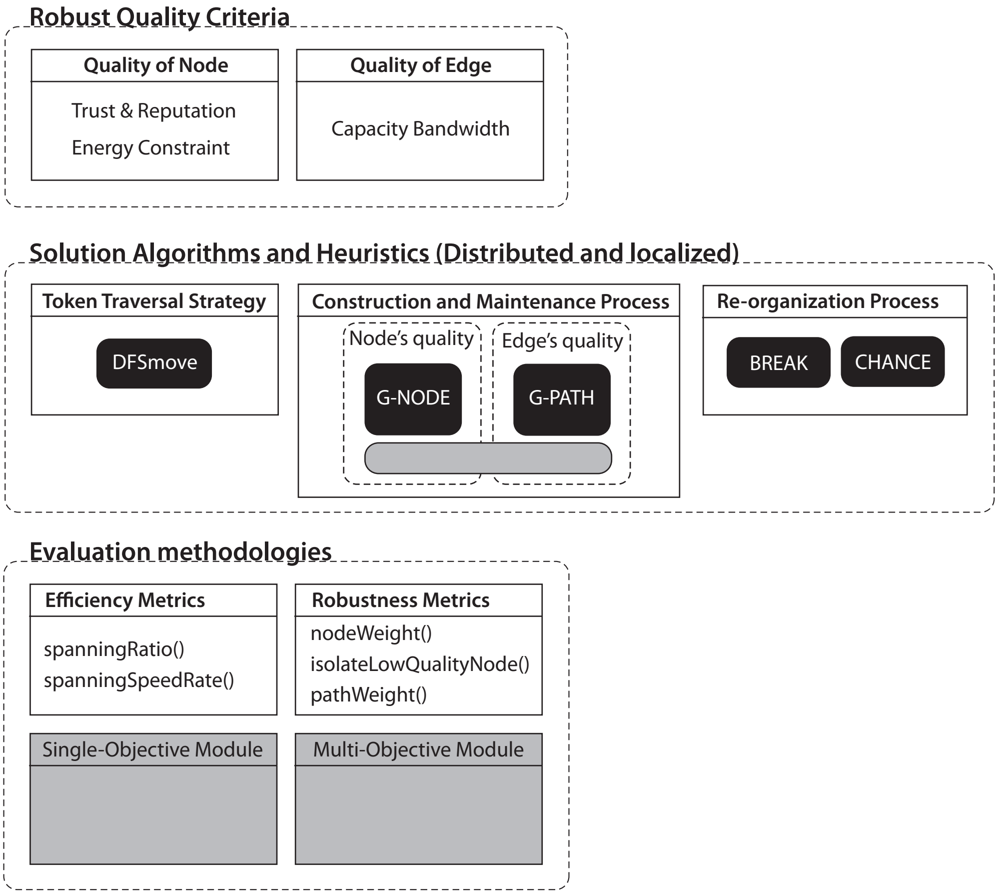


Figure 4.18: Framework proposed by this study including construction, maintenance and re-organization process

So far, we separately addressed problems of robust spanning forest topology based on criteria of node and edge. However, it is more suitable to address the problem considering multi-criteria or at least bi-criteria at the same time. Thus, we propose the multi-objective approach to solve robust spanning forest problem in DT-MANETs. This section comprises of the proposal algorithms to solve the problem in decentralized and asynchronous manner while using only one-hop information. In order to validate the result, our framework provides comparison between the previous algorithm (DA-GRS reference algorithm), our proposed algorithm and the optimal value. The proof of optimality for each objective are also provided in this chapter. The framework integrates jMetal [126] and proper set of methodologies for finding Pareto front of multi-objectives problem.

4.7.1 Problem Description

Different qualities of nodes and their communication links are used as robust criteria to topology management in DT-MANETs problem. This work proposes a framework to provide robust topology to DT-MANETs' topology management problem by considering multi-criteria at the same time. Spanning Forest is used as an example of desired topology. Robustness objectives of such topology when considering both node and link qualities are described in 4.4.1 and 4.4.3 respectively. The most basic problem, thus, is addressed as bi-objectives problem, considering one criteria of node's quality and one criteria of edge's quality. By having conflicting quality criteria, a bi-objectives problem according to our interest is described formally as follow:

Given $G(t) = (V(t), E(t))$ a communication graph at moment t and γ a tree on $G(t)$. The problem of robust spanning forest topology in DT-MANETs is:

$$\begin{aligned} &maximize \quad nodeWeight(G(t)) = \sum_{\forall \gamma \in G(t)} nodeWeight(\gamma) \\ &maximize \quad pathWeight(G(t)) = \sum_{\forall \gamma \in G(t)} pathWeight(\gamma) \end{aligned}$$

where $nodeWeight(\gamma)$ is defined in equation 4.4 and $pathWeight(\gamma)$ is at equation 4.8. However, for ease of understanding we state them again here:

$$nodeWeight(\gamma) = \sum_{x \in V(\gamma)} quality(x) \times tree\_degree(x)$$

$$pathWeight(\gamma) = \sum_{\forall (a,b) \in E(\gamma)} M_{CB} \; P(a,b)$$

It must be emphasized here that for multi-objective problems, with several conflicting objectives, there is usually no single optimal solution. MOOP presents a possibly uncountable set of solutions, which when evaluated, produce vectors whose components represent trade-offs in objective space. Therefore, the decision maker is required to select a solution (or solutions) from a finite set by selecting one or more of these vectors and making compromises. To the problem of managing topology in DT-MANETs like this one, the proposed algorithm must have ability to provide such a variety set of solution vectors: leave the decision on which compromising level is proper at the moment to the application level.

4.7.2 Proposition of a solution: 'G-Node-Path'

'G-Node-Path' algorithm has its base on the composition of G-NODE (4.6.2.1) and G-PATH (4.6.2.2) described earlier. It takes into account the compromising issue and give robust spanning forest solutions according to the compromise setting. Moreover, 'G-Node-Path' works in fully decentralized and asynchronous manner, which is suit to the characteristics of DT-MANETs. Indeed, the algorithm makes decision locally and in discrete manner. The preference of trade-off value is denoted as $[\rho, 1-\rho]$ while ρ is preference value attached to one of the objective and the other objective will logically have preference value of $1-\rho$. This preference value is used to influent the selection of used algorithm, G-NODE or G-PATH, at one moment of each node. By discretely using ρ to induce the selection of algorithms, the ultimate goal is to control the global behavior of managed spanning forest, such that robust topology follows the preference characteristics.

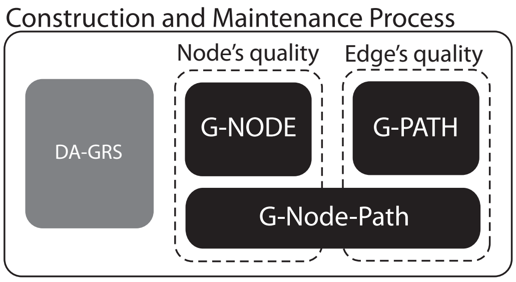


Figure 4.19: Relationship between G-NODE, G-PATH and G-Node-Path

4.7.3 Optimum solution to single-objective function

4.7.3.1 Reduction of $nodeWeight(\gamma)$ objective

Given $G = (V, E)$ a communication graph
and γ a tree on $G(t)$
$\gamma = (V, E_\gamma)$
For each vertex v_i we note w_i its battery level
d_i its degree in γ
For a given γ we consider the following metric:

$$f_\gamma = \sum_{v_i \in V} w_i \times d_i^\gamma$$

Problem: We are looking for γ^* such that $f_{\gamma^*} = \max f_\gamma$

Proposition: We claim that maximizing f for G is equivalent to find a maximum spanning tree for G.

Proof: For one particular spanning tree γ:

$$\begin{aligned} f_{\gamma^*} &= \sum_{v_i \in V} w_i \times d_i^\gamma \\ &= \sum_{v_j \in N(v_1)} (w_1 + w_j) + \sum_{v_i \in V \setminus \{v_1\} \setminus \{N(v_1)\}} w_i \times d_i^\gamma + \sum_{v_k \in N(v_1)} w_k \times (d_k^\gamma - 1) \end{aligned}$$

where $N(v_1)$ is the set of neighbors of v_1.

$$\begin{aligned} &= \sum_{v_j \in N(v_1)} (w_1 + w_j) + \sum_{v_l \in N(v_2)} (w_2 + w_l) + \sum_{v_i \in V \setminus \{v_1, v_2, N(v_1), N(v_2)\}} w_i \times d_i^\gamma \\ &\quad + \sum_{v_k \in N(v_1) \setminus N(v_2)} w_k \times (d_k^\gamma - 1) + \sum_{v_m \in N(v_2) \setminus N(v_1)} w_m \times (d_m^\gamma - 1) \\ &\quad + \sum_{v_n \in \{N(v_1) \cap N(v_2)\}} w_n \times (d_n^\gamma - 2) \end{aligned}$$

We continue the same transformation and it comes that

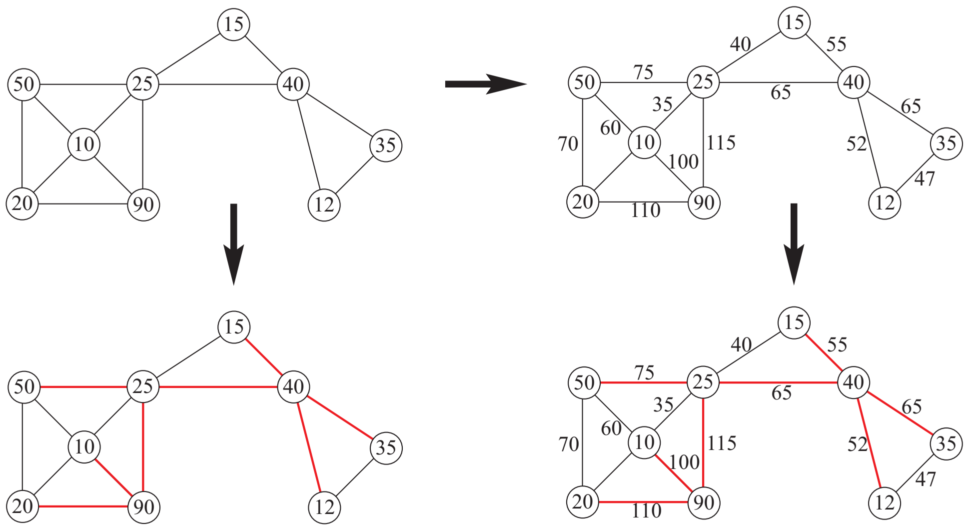
$$\sum_{v_i \in V} w_i \times d_i^\gamma = \sum_{(v_i, v_j) \in E_\gamma} (w_i + w_j)$$

Thus,

$$\max \sum_{v_i \in V} w_i \times d_i^\gamma = \max \sum_{(v_i, v_j) \in E_\gamma} (w_i + w_j)$$

and $\max \sum_{(v_i, v_j) \in E_\gamma} (w_i + w_j)$ is obtained for the maximum spanning tree for G where each edge is evaluated by the sum of the battery level of its adjacent nodes.

Example (illustration): The example show below illustrates the equivalent of our claim that maximizing f for G is equivalent to find a maximum spanning tree for G. The resulting spanning tree is depicted using red line and the optimal value of that tree is 647.



(a) Robust spanning forest topology problem (b) Maximum spanning tree problem

Figure 4.20: An example illustrates the equivalent of robust spanning forest topology problem to maximum spanning tree

Summary: The single objective of $nodeWeight() function$ problem can be transformed into maximum spanning tree problem. This is done by giving value on edge using an addition of adjacent nodes' battery level. Thus, this single objective problem is able to find optimum value using either Kruskal or Prim's algorithm when the global information is allowed.

4.7.3.2 Reduction of $pathWeight(\gamma)$ objective

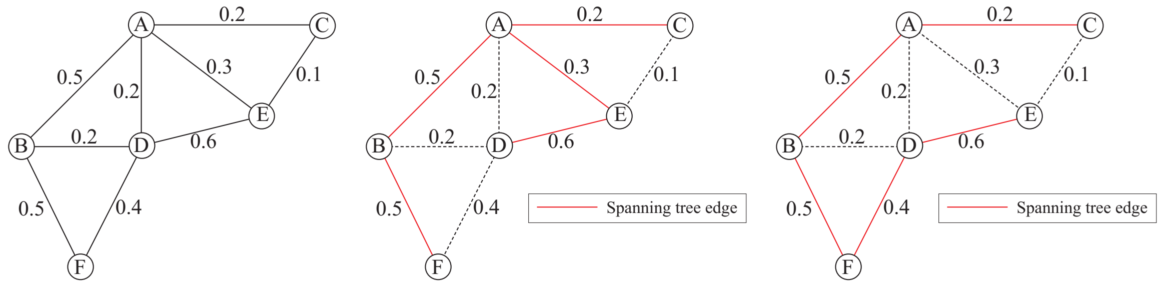
Given	$G = (V, E)$ a communication graph
	and γ a tree on $G(t)$
	$\gamma = (V, E_\gamma)$
For each edge $(v_i, v_j) \in E_\gamma$ we note	$CB_{(v_i,v_j)}$ its capacity bandwidth (Def: 45)
For each Path exists in γ, $P(v_s, v_d)$ we note	$M_{CB}P(v_s, v_d)$ as the minimum value of
	the CB of any edge belonging to $P(v_s, v_d)$
For a given γ	we consider the following metric:

$$f_{\gamma^*} = \max_{\forall (a,b) \in E(\gamma)} (M_{CB}\ P(a, b))$$

Problem: We are looking for γ^* such that $f_{\gamma^*} = \max f_\gamma$

Proposition: We claim that maximizing f for G is equivalent to find a maximum spanning tree for G.

Proof: Considering graph G and one particular spanning tree γ_1 shown below:



(a) Graph G

(b) one particular spanning tree γ_1

(c) Maximum Spanning tree of graph G

Figure 4.21: Example graph and spanning tree for reduction pathWeight() function proof

Consider a path of (v_x, v_y), where $\exists\ P(v_x, v_y) \notin$ spanning tree γ_1 such that M_{CB} is greater than in γ_1. If this is the case, then there is an edge $(v_i, v_j) \notin \gamma_1$. We denote a spanning tree γ_2 contains such path, $P(v_x, v_y)$, which contains an edge (v_i, v_j) such that

$$\sum_{\forall (v_x,v_y) \in E(\gamma_2)} M_{CB}\ P(v_x, v_y) > \sum_{\forall (a,b) \in E(\gamma_1)} M_{CB}\ P(a, b)$$

and

$$M_{CB}(v_i, v_j) > M_{CB}(w, z), where(w, z) = E_{\gamma_1} \setminus E_{\gamma_2} \setminus (v_i, v_j).$$

Thus, only the maximum M_{CB} of $|V|-1$ edges of graph G must contains in the spanning tree that give maximizing f for G. Hence, $\max\limits_{\forall(a,b)\in E(\gamma)} (M_{CB}\ P(a,b))$ is the same topology set of spanning tree, edge set, as the maximum spanning tree for G. Thus, v_i, v_j is edge (D, F) found in Sub-Figure 4.21(b). Figure 4.21(c) illustrates the maximum spanning tree to example graph G in Sub-Figure 4.21(a).

Example (illustration): The example show below illustrates the equivalent of our claim that maximizing f for G is equivalent to find a maximum spanning tree for G. The resulting spanning tree is depicted using red line. Although both figure show the same spanning tree, the optimal value obtained are different according to the pathWeight() function. However, it is guaranteed the maximization of the solution.

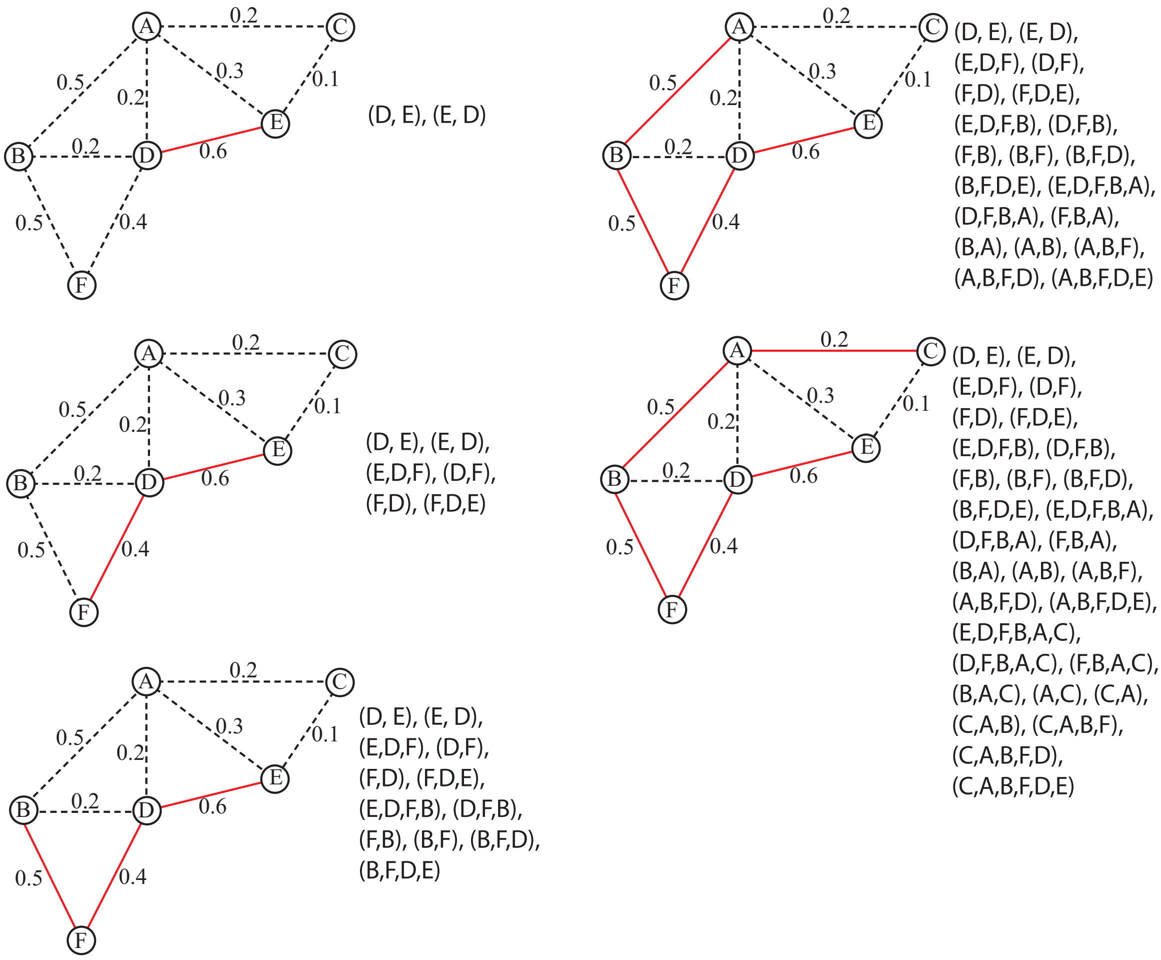


Figure 4.22: example of max pathWeight() spanning tree

Summary: The single objective of pathWeight()function problem can be transformed into maximum spanning tree problem. This is done by finding the maximum spanning tree based on Capacity Bandwidth value on each edge. Thus, this single objective problem is able to find optimum spanning tree using either Kruskal's or Prim's algorithm when the global information is allowed. However, the optimal value to the function pathWeight() differ from the summation of minimum capacity bandwidth of all possible path from the maximum spanning tree. The optimum solution of pathWeight() function can be found by calculated from the resulting maximum spanning tree.

4.7.4 Optimal solution to multi-objective function

4.7.4.1 Enumerating the whole set of spanning trees: Exact Approach

Cayley [127] proved that for a complete graph K_n, where n is number of nodes, there are exactly n^{n-2} possible trees. For example, if we have a complete graph K_{10}, the number of non-identical spanning tree is 10^8. Please note that for any complete graph K_n, there exists $n(n-1)/2$ edges. Let look at another example, given a complete graph of 52 vertices, which means it exists 1,326 edges, the number of non-identical spanning trees accordingly is 6.31×10^{85}.

For any non-complete graph G_{nc}, Kirchoff's theorem or the matrix-tree theorem [128] can give the number of non-identical spanning trees of any G_{nc}. This result uses the $p \times p$ degree matrix $C = [c_{ij}]$ of G_{nc}, where $c_{ij} = deg\ v_i$ and $c_{ij} = 0$ if $i \neq j$. This result is known as the matrix-tree theorem. For each pair (i, j), let the matrix B_{ij} be the $n-1 \times n-1$ matrix obtained from the $n \times n$ matrix B by deleting row i and column j. Then $det\ B_{ij}$ is called the minor of B at position (i, j) and, $(-1)^{i+j}\ det\ B_{ij}$ is called the cofactor of B at position (i, j).

Definition 47 (The Matrix-Tree Theorem).
Let G be a nontrivial graph with adjacency matrix A and degree matrix D. Then the number of non-identical spanning trees of G is the value of any cofactor of D - A. [129]

According to the theorem 47, the topology of any non-complete graph G_{nc} has an influence on the number of non-identical spanning trees of G_{nc}. A connected graph of $n-1$ edges has the topology of spanning tree, and thus, it is the lowest bound: there is only one tree. On the other hand, the upper bound is found in complete graph; there exists $n(n-1)/2$ edges and n^{n-2} trees. This means it is plausible to enumerate the whole set of spanning trees when the number of nodes in considering graph is small. However, it is a non-trivial task to give all spanning trees for a not so small number of nodes. For this reason, a empirical study has been launch in the following.

Shioura [130] proposed an algorithm to find all possible set of spanning trees using only O(N+V+E) when N is the number of all possible set of spanning trees. With this boundary, it seems to give a polynomial time and space. However, the factor of N is exponential in calculation. Thus, It is non-trivial in finding all spanning trees for a graph with high number of nodes and edges. To proof this, an experimentation is conducted by a machine with the following details: the processor is 2×2.66 GHz Dual-Core Intel Xeon, the memory is 8 GB 667 DDR2 RAM, the L2 Cache is 4 MB. We consider this machine is a powerful one at the experimenting time. In this experiment, each connected component set is a connected but not a complete graph. However, only small number of vertices, lower than 52 nodes, are successfully used to enumerate all spanning trees using algorithms of [130]. Table 4.4 reports the properties of each connected component (i.e. number of nodes and edges), number of spanning trees and the simulation result. In order to demonstrate time and space used, instead, we use four set of small complete graphs shown in Table 4.5.

Table 4.4: Result of Enumerating Whole Set of Spanning Trees using Different Non-Complete Graph and Size of Vertices

Connected Component Set	Number of Nodes	Number of Edges	Number of Spanning Trees	Retrieve all Spanning Trees
1	9	21	27,440	Yes
2	16	40	3.34×10^7	Yes
3	52	248	6.04×10^{43}	No
4	87	655	2.69×10^{94}	No

Table 4.5: Enumerating all Spanning Trees Results from Complete graphs

Test Set	No. of Nodes	No. of Edges	No. of Trees	CPU Times	Elapse Times (second)	Memory Used	data Storage
1	6	15	1.3×10^3	3	0.5	36 MB	5.1 MB
2	7	21	1.7×10^4	31	12	43 MB	66 MB
3	8	28	2.6×10^5	543	270	113 MB	1.00 GB
4	9	36	4.8×10^6	10,676	6,120	1.35 GB	18.21 GB
(*CPU times obtained by function *tms_utime* of library sys/times.h. The system measures time by counting clock interrupts.)							

This empirical study confirms that enumerating all spanning trees is a non-trivial task. The number of all possible spanning trees is exponential if the considering number of nodes is not very small. Since the task of solving multi-objective optimization is to find optimal solutions, in this case the best solution of tree must be found among large possibility choices. In order to avoid combinatorial explosion in finding optimal solutions for multi-objective spanning tree problems, the meta-heuristic approach is more suitable.

4.7.4.2 Meta-Heuristics Approach

Introduction: Evolutionary algorithms (EAs) are very popular approaches in multi-objective optimization. EAs mimic natural evolutionary principles to constitute search and optimization procedures. Finding and maintaining multiple solutions in one single simulation run is a unique feature of evolutionary optimization techniques. In the trajectory based method, it does a complete search and compare a single solution in every iteration. This fact make use a huge computation power and time. On the other hand, MOEA tends to give a faster solution by using population-based solution in every iteration.

'jMetal' [126] is an object-oriented Java-based framework aimed at the development, experimentation, and study of meta-heuristics for solving multi-objective optimization problems. jMetal provides a rich set of classes which support the implementation of a new meta-heuristic and a new application problem. In this work, we focus to our own problem which is about finding optimal spanning forest or a set of spanning trees, according to our own robustness metrics as shown in the previous section 4.7.1.

Representation of Spanning Tree in Evolutionary Algorithm Working with EAs, a spanning tree problem needs to encode a solution (tree) such that evolutionary search operators like crossover or mutation can be applied. There are two different approaches for doing this: indirect representations and direct representations. The former approach usually encode a tree (phenotype) as a list of strings (genotypes) and apply standard search operators to the genotypes. The phenotype is constructed by an appropriate genotype-phenotype mapping. There are many indirect representations for trees such as the Prüfer numbers [127], Character vector [131], Blob Code [132], [133], NetKeys [134] etc.

In contrast, direct representations encode a tree as a set of edges and apply search operators directly to the set of edges. Therefore, mapping is not necessary. Instead, tree-specific search operators must be developed as standard search operators can no longer be used. Since there is no additional genotype-phenotype mapping, here tree-specific search operators are directly applied to the phenotypes.

In this work, we use Edge-Set [135] proposed by Raidl and Julstrom in 2003 as representation of spanning trees. Not only the direct representation, Julstrom [135] also provides some idea of initialization, crossover, and mutation operators for working with Edge-Set too. Our tasks was to implement the idea and fitting them into jMetal framework.

4.8 Summary

This work aims at providing efficient and robust topology in DT-MANETs by means of distributed, asynchronous, and localized heuristics. Utilizing robustness qualities found on node and edge of communication network are proposed and discussed under this Chapter. The robust quality we mentioned in this work are, such as, trust value from cooperative enforcement paradigm, battery level from energy availability approach, and capacity bandwidth regarding end-to-end communication. These robust qualities can be selected to use for a single objective problem or for a multi-objective problem.

Since different desire of being robust may differ from one situation to another, this study provide solutions in terms of framework where each module can be seen as plugin. The framework comprises of (1) a set of robust criteria for topology management problem, (2) the evaluation methodologies for different solutions of different problems (e.g., single-objective and multi-objective problem). and (3) a set of distributed algorithms/heuristics as solutions providing robust spanning forest in DT-MANETs. The framework also provide the procedure to obtain optimal value for both single-objective and multiple-objective problem. This study focuses on spanning forest as a desire robust topology in DT-MANETs. Accordingly, nodeWeight(), isolateLowQualityNode() and pathWeight() functions are proposed to evaluate the quality of constructed spanning trees. Figure 4.23 summarizes the framework proposed by this study.

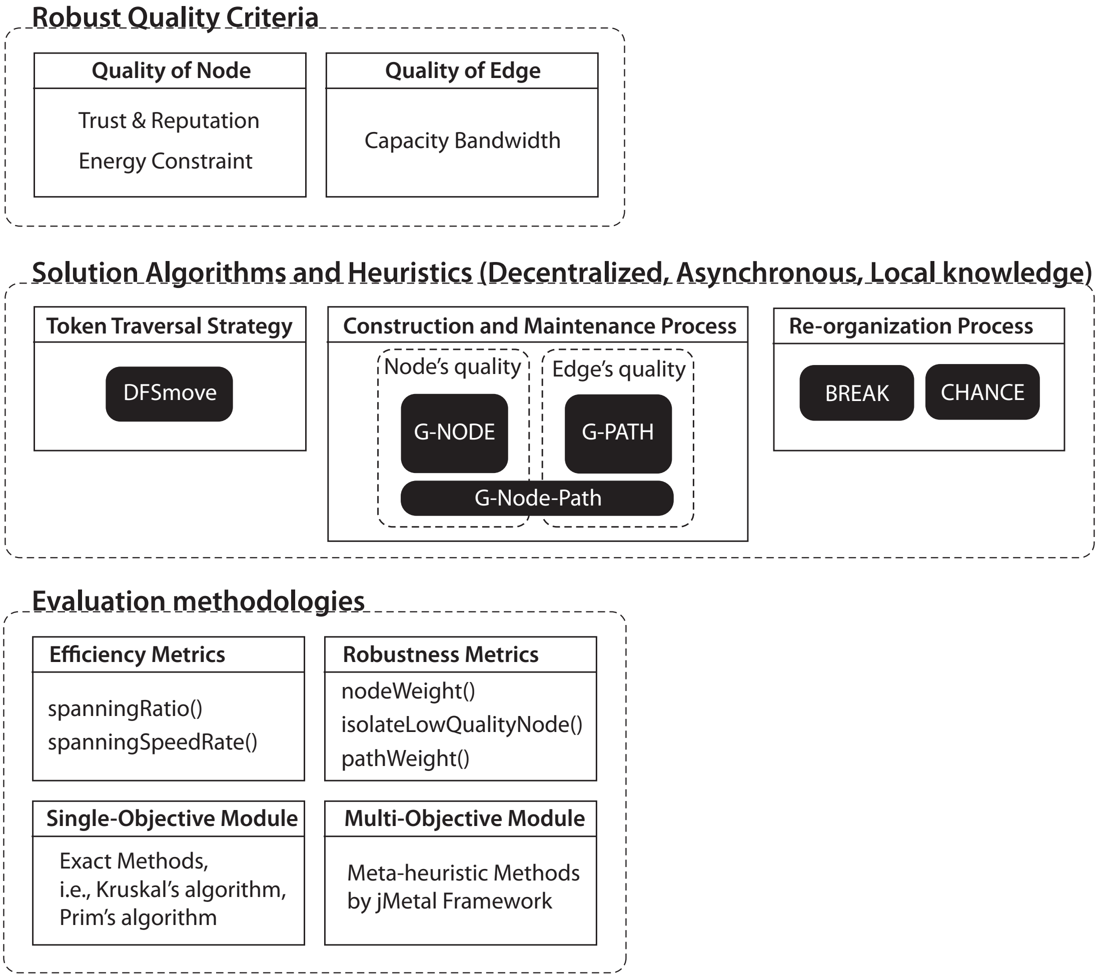


Figure 4.23: Proposed Framework for Topology Management in DT-MANETs based on Spanning Forest

Every module in the framework are plugin where the composition between them is possible. For example, the desire robust topology at one moment is to have high capacity bandwidth. The capacity bandwidth is used as robust quality criteria. Algorithm G-PATH is selected to construct robust spanning forest because the desire robustness is based on edge's quality. Our DFSmove is selected as the token traversal strategy and, if needed, BREAK or CHANCE re-organization process may be selected.

All proposed distributed, asynchronous, and localized algorithms are evaluated by simulation in an online phase. Within this phase, the proposed algorithms are installed into each node prior to the simulation. The simulating is done in realistic manner, that is no global knowledge is assumed and using realistic mobility models at different configuration and topology. The results obtained were evaluated according to the problem description in an offline phase. In this offline phase, global information is assumed. The single-objective problem in this study was reduced to a similar well-known problem (maximum spanning tree problem) with some special treatment. Thus, optimal topology solution can be obtained in polynomial time, using Kruskal's or Prim's algorithm. This has been proved and given examples under sub-section 4.7.3 for both objectives based on node's and edge's quality. On the other hand, optimal solutions of multi-objective problem were found by multi-objective optimizer, jMetal with NSGA-II metaheuristic.

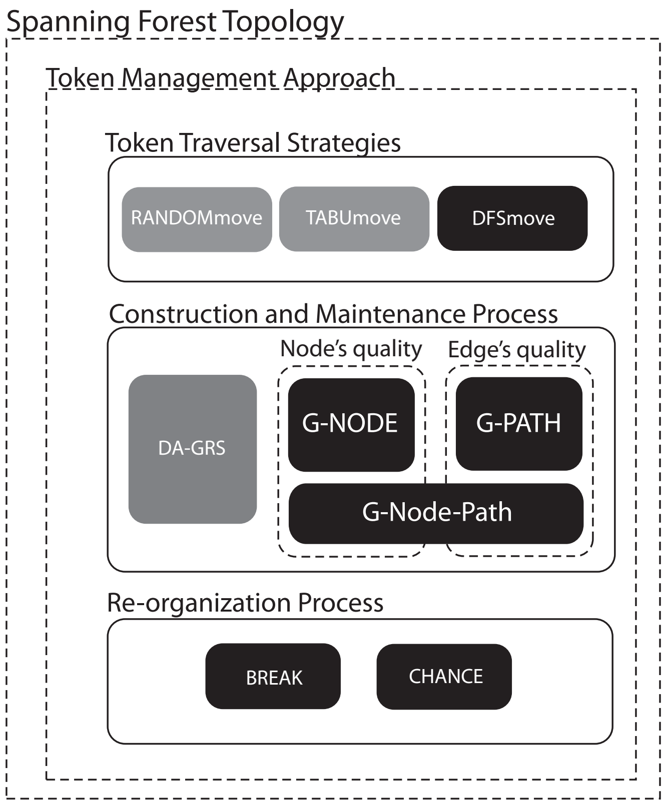


Figure 4.24: Purely decentralized and one-hop information used heuristics proposed by this work

The algorithms and heuristics propose for online phase in this work can be summarized in Figure 4.24. Existing techniques are in grey tones, while new algorithms and heuristics implemented in this study are shown in black tones. As stated earlier, this work focuses on finding robust spanning forest topology and there are several ways to obtain spanning tree in literature. According to constraint of DT-MANETs, we need a purely distributed algorithm with the very local information usage. Our approach concentrated on token management strategy for creating spanning tree provided by DA-GRS model. We improve the token traversal strategies by cooperating DFSmove into DA-GRS. Furthermore, we introduce realistic quality value on node and edge which was absent in the original DA-GRS. G-NODE, which is a purely decentralized and influences by greedy algorithm, are proposed to construct spanning forest according to node's quality. Similarly, G-PATH is implemented to construct spanning forest according to edge's quality. From the multi-objective scheme, the research introduces G-Node-Path as a tool to combine both G-NODE and G-PATH. It finds a compromise between the two original heuristics in order to obtain the final spanning forest according to the problem description.

However, this work also considers a wide range of mobility (i.e., ranging from human walking to cars running on highway). As the dynamic of the network constantly changes, the robustness of the spanning forest may deteriorate. In order to overcome, thus, the maintenance of spanning forest is improved by means of BREAK and CHANCE heuristics to re-organizing the structure. The next chapter discusses the results of all new techniques and proposal discussed in this work.

Chapter 5

Experimentation and Validation Methodologies

Contents

5.1 Experimental Setup 99

5.1.1 Used Simulators 99

5.1.2 Analysis of Used Mobility models 100

5.1.3 Assumptions 107

5.1.3.1 datagram used in this work 107

5.1.3.2 Communication Assumptions in GraphStream 107

5.2 Results and Analysis of the Proposal 111

5.2.1 Tree Traversal Strategies 111

5.2.1.1 Preliminary 111

5.2.1.2 Tree Traversal Strategies over DA-GRS reference algorithm 111

5.2.1.3 Tree Traversal Strategies over the Robustness of Trusted Tree Topology by G-NODE 114

5.2.2 Robust topology utilizing Cooperative Enforcement Approach 118

5.2.3 Robust topology utilizing Energy Availability Approach 121

5.2.4 Robust topology utilizing Capacity Bandwidth Approach 124

5.2.5 Reorganization of topology by BREAK heuristic 127

5.2.6 Reorganization of topology by CHANCE heuristic 130

5.2.7 Efficient and Robust Topology in Multi-Objective Optimization Approach 134

5.2.7.1 Experimentation methodology and Implementation note . . 134

5.2.7.2 Hypervolume results 137

5.2.7.3 Robustness Metrics Results 138

5.3 Summary 139

All the underlying method of experimentations and validation has been described and declared in this Chapter. The extensive analysis of the different dynamic network are provided. Some special setting and/or the detailed characteristic of used network for each results are introduced. The results are then provided accordingly.

5.1 Experimental Setup

In ad hoc wireless network research area, it is very costly to deploy a complete testbed containing multiple wireless devices to validate and verify a certain network protocol or a specific network algorithm. The simulation methodology can save a lot of money and time in accomplishing those tasks. Moreover, simulators are useful in the sense that it allows the testing and changing the studying protocols and algorithms in a controlled and reproducible manner. Thus, this study uses simulation as a mean for evaluation our proposed algorithms and heuristics.

5.1.1 Used Simulators

This study particularly interested in human mobility model (HMM) of [136]. As discussed earlier under section 2.1.4.2, human mobility model is a generic mobility model that represents the intention-driven mobility of people in metropolitan areas and also provides different scenario modeling such as Mall and Highway mobility models. Although, random waypoint model is a very common mobility model but its operations is based on randomness. Therefore, its nature of a pure mathematical model makes this unsuitable for human mobility. Hogie [40] provides a network simulators named 'Madhoc' equipped with a set of human mobility model (HMM) generator. To be more specific, shopping mall and highway mobilities are selected for this study due to their strong and different characteristics. More detail on mobility models used in this work is discussed in the next sub-section.

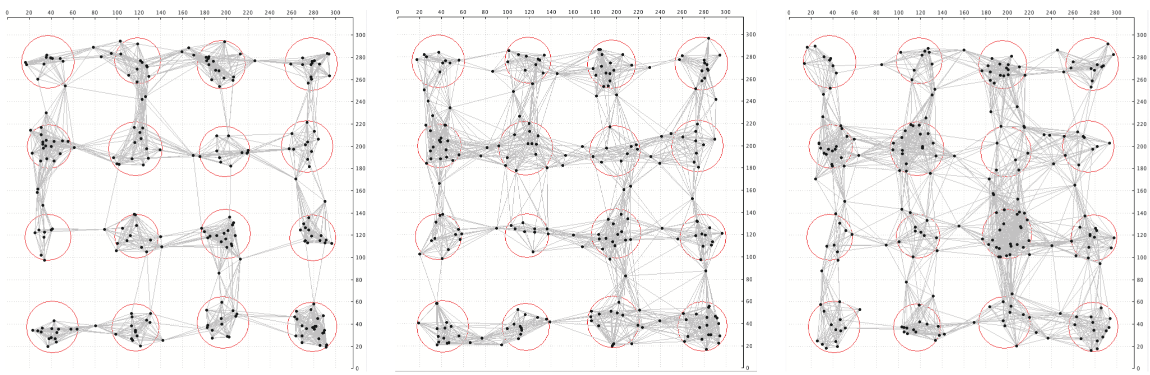
Unlike many other simulators, Madhoc does not rely on the discrete-event simulation paradigm. Instead of jumping from event to event, Madhoc's kernel iterates upon time and time is discrete. Madhoc simulates MANET according to a set of pre-specified data such as time-interval between two iterations, types of communication technologies on mobile nodes, density of mobile nodes, simulation area, mobility models. The different mobility models provided by Madhoc allow realistic motion of citizens in variety of environments. Madhoc also provides trace files of each simulation. Thus, these trace files capture all the event of network according to the set of pre-defined data. These events are 'adding node', 'adding edge', 'removing node' and 'removing edge' at each time step t_i. According to a specified time interval given by the pre-defined data, list of event at each time step t_i is the result of event happened during the interval t_{i-1} and t_i.

Madhoc is a network simulator with ability to provide realistic mobility model. Unfortunately, its underlying stochastic model would not allow repeating exactly the same scenario. In Madhoc, we generate realistic trace files and then use and replay them in a deterministic way using GraphStream [137]. GraphStream is a java library that manages dynamic graphs. It is composed of an object oriented API that gives a quick and easy way to add edges and nodes in a graph in order to evolve them. Utilizing trace files from Madhoc in GraphStream is easy, since the syntax of trace file is compatible between two platforms. All of our simulations are done based on GraphStream simulator.

5.1.2 Analysis of Used Mobility models

Mobility of nodes is very important factor affecting the performance (throughput) of networking protocol. As discussed in Chapter 2 nodes in DT-MANETs are assumed to be mobile type and having movement most of the time. Information on bandwidth reservations or the control information exchanged may become outdated and obsolete if the node mobility is very high. Under this section, we shall discuss in particular about the mobility models we used for this study.

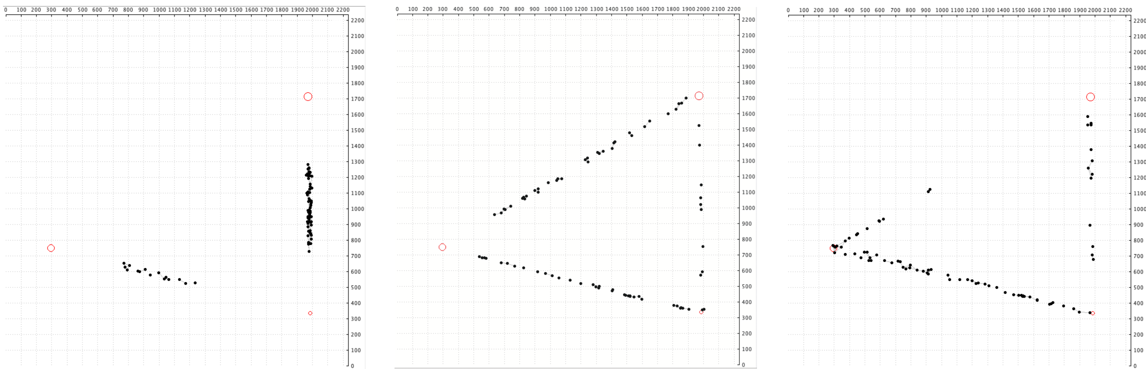
Two real-world mobility models, 'Shopping Mall' and 'Highway', were selected for the simulations. They were chosen for this research as their characteristics are almost the opposite of each other, and thus, we could use them to ensure the validity and robustness of the verification. Shopping Mall presents a mobility model that nodes are moving based on walking to running speed. Within a shopping mall area, there exists several spots representing different shops. Inside shops, the density of mobile nodes can be expected higher than in the corridor area. The direction of nodes in shopping mall model can have high fluctuation since there is no specific rule of walking. In contrary, the highway mobility model shows us the speedy movement of cars on highway but nodes move with guidance of road infrastructure. An important characteristic of highway model is the opposite movement of mobile nodes due to the two different directions of road infrastructure. Although, highway mobility model presents high speed of nodes, one common characteristic is also the group of node has same sped and move in the same direction. These nodes become neighbors to each other for a period of time before any node change their speed and direction. Some snapshots of the obtained mobility model network from Madhoc are shown in Figure 5.1 and 5.2 for Shopping Mall model and Highway model respectively.



(a) A Snapshot at G(1) (b) A Snapshot at G(200) (c) A Snapshot at G(400)

Figure 5.1: An example of Shopping Mall Mobility Model with 240 nodes

We derived communication graphs from Madhoc which performs simulation in discrete-time. So the communication network corresponds to a series of static graphs: $G(t)$ for $t \in \{t_1, t_2, t_3, ..., t_{400}\}$. Between two consecutive times t_i and t_{i+1} the communication graph remains the same. A short timing-snapshot, 1/4 seconds between two consecutive times is



(a) A Snapshot at G(1) (b) A Snapshot at G(200) (c) A Snapshot at G(400)

Figure 5.2: An Example of Highway Mobility Model with 80 nodes

considered sufficient to reflect the reality. As can be seen in Figure 5.1 and 5.2, differences in dynamicity of snapshots are shown in their sub-figures at different $G(t)$.

Four configurations were generated in this study. The parameters of these four different models are given in Tables 5.1 and 5.2 for Shopping Mall and Highway mobility model networks respectively. These set of networks are used in some experimentations which have been crossed referenced with the experimentation results in Section 5.2. Furthermore, the corresponding topology of each configurations can be found in Figure 5.3(a) and 5.3(b) for all shopping mall and highway configurations respectively.

Table 5.1: Parameters used in the experiments for Shopping Mall mobility model

	Configuration			
	1	**2**	**3**	**4**
Surface (km^2)	0.32			
Node Density (per km^2)	250	500	750	1000
Number of Nodes	80	160	240	320
Avg. Number of Partitions	3.48	1.16	1.00	1.00
Number of Connections	219.79	970.27	2121.83	3772.72
Average Degrees	5.51	12.16	17.72	23.63
Velocity of Nodes (m/s)	0.3-3			
Radio Transmission Range	40-80 m			
Network Technology	IEEE802.11b			

The differences of these configurations are mainly the number of nodes and edges which are effected by the density, as shown in Table 5.1 and 5.2. Furthermore, the number of connected components is also influenced by the density factor too. The Figure 5.3(a) and 5.3(b) show the dynamic number of edges and connected components over the simulation time for both mobility models.

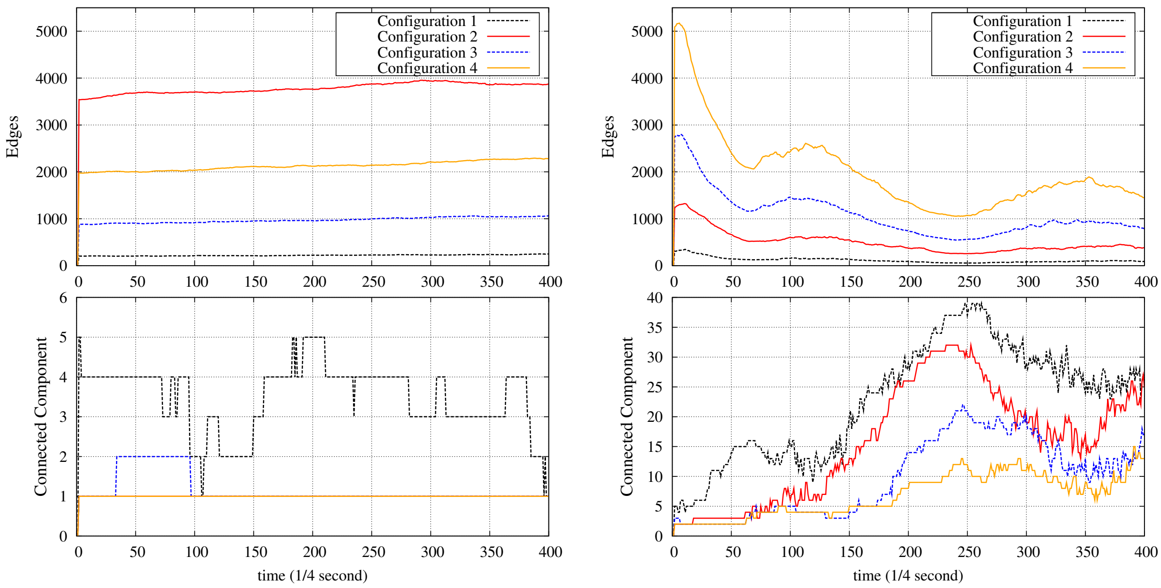
Considering age of all edges in each configuration, extensive analysis to the simulated networks were carried out. Especially, when deals with different mobility models like in this

Table 5.2: Parameters used in the experiments for Highway mobility model

	Configuration			
	1	**2**	**3**	**4**
Surface (km^2)	2.24			
Node Density (per km^2)	35.72	71.43	107.14	142.86
Number of Nodes	80	160	240	320
Avg. Number of Partitions	22.78	15.90	9.89	6.91
Number of Connections	117.59	498.00	1078.00	1983.04
Average Degrees	2.96	6.23	9.04	12.47
Velocity of Nodes (m/s)	20-40			
Radio Transmission Range	40-80 m			
Network Technology	IEEE802.11b			

study, age of edges per simulation step shows the stability of all possible communication links. This study uses box plots (also known as 'box and whisker plots' or 'boxplot') to illustrate the age of all edges at each simulation step. In descriptive statistics, a box plot is a convenient way of graphically depicting groups of numerical data through their five-statistical summaries: the smallest observation (sample minimum), lower quartile (Q1), median (Q2), upper quartile (Q3), and largest observation (sample maximum). The interquartile range or IQR is the differences between the upper quartile (Q3) and lower quartile (Q1). IQR is a robust way of describing the dispersion of the data. It shows the range between Q1 and Q3. Box plots are uniform in their use of the box: the bottom and top of the box are always the 25th and 75th percentile (Q1 and Q3, respectively), and the band shown within the box is always the 50th percentile (the median: Q2). However, the ends of the box plot (representing with lines) can represent several possible alternative values. In this study, we use the lowest datum still within 1.5 IQR of the lower quartile, and the highest datum still within 1.5 IQR of the upper quartile as the ends of the box. The information falls outside this end are extreme value or outliers.

Figure 5.4 gives an example of box plot. The red box presents the lower and the upper quartile (Q1 and Q3 respectively). In general, there exist further lines from both ends of the box except for information at point 7 and 8 of x-axis. Both of them have only one further line either at the lower side or the upper side. At point 7, there is no further line below the quartile box. This means the value at the 25th percentile is at 4, the IQR at this point is 2 and the lowest datum within 1.5 IQR of the Q1 is at 4. These imply that there exist a large number of value 4 in the data set, but not exceed 50% of the data (median is at 4.8). On the other hand, the data at point 8 of x-axis shows only the median point without the visible of quartile red box. Actually, the value of Q1, Q2 and Q3 are all the same at value 5. This makes IQR value equals to zero and thus there is no value of 1.5 IQR. However, we cannot summarize that there is no explicit lowest and the highest value at this point. Likewise, the rest of box plots in this example do not show the minimum and the maximum value of all point since the value of 1.5 IQR is used to present the boundary of the data (at 95% of distribution). In our study, the minimum and maximum value are also interesting and will be added into the graph using separate lines.



(a) Shopping Mall Topology

(b) Highway Topology

Figure 5.3: Number of Edges and Connected Component of each configurations from both Mobility Model according to Table 5.1 and 5.2

Figure 5.5 and 5.6 illustrate the age of all edges from both highway and shopping mall mobility models according to each configuration shown in Table 5.1 and 5.2 respectively. In these figures, the median of age value among edges is shown using red color bars, and the minimum age is presented by blue line. The maximum age value is not taking care explicitly in this analysis because the data set tend to reach the maximum value as well as the median of the data. Taking an example from sub-figure 5.5(a), we immediately see that the median at each step is always the number of simulation step itself until approximately step 335, where the median value decreases. This means before the step 335 more than half of all edges has its age equal to the simulation step and after the step 335 these number of long live edges has been decreased and less than half of all edges. Furthermore, before the step 160, the box has been plotted under the median which means the median value has population covering both Q1 and Q3. Thus, the IQR equals to zero. This also implies that the number edges having different age value is very low (less than 25%). As shown in color blue line, the other edges with other age value are exist but at an extremely small number, and thus, the stability of edges is quite high. Between step 160 and 335, the wide range of age information are shown, but the median still insist at the maximum age. This implies that there exist multiple edges existing in the network some times after the initialization and continue to exist in the network. At the same time, some very old ages cease but still hold the majority of the edge population (more than 50% of population but less than 75%).

For highway mobility model, the fluctuating of edges' appearance and disappearance can be observed directly. Also, there exists the duration time where age of edges are higher than

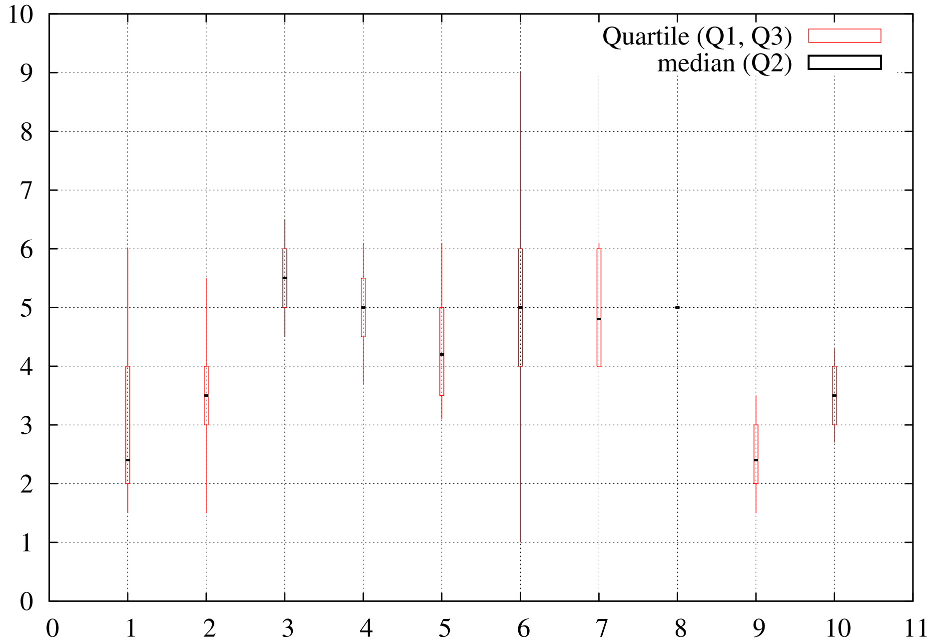
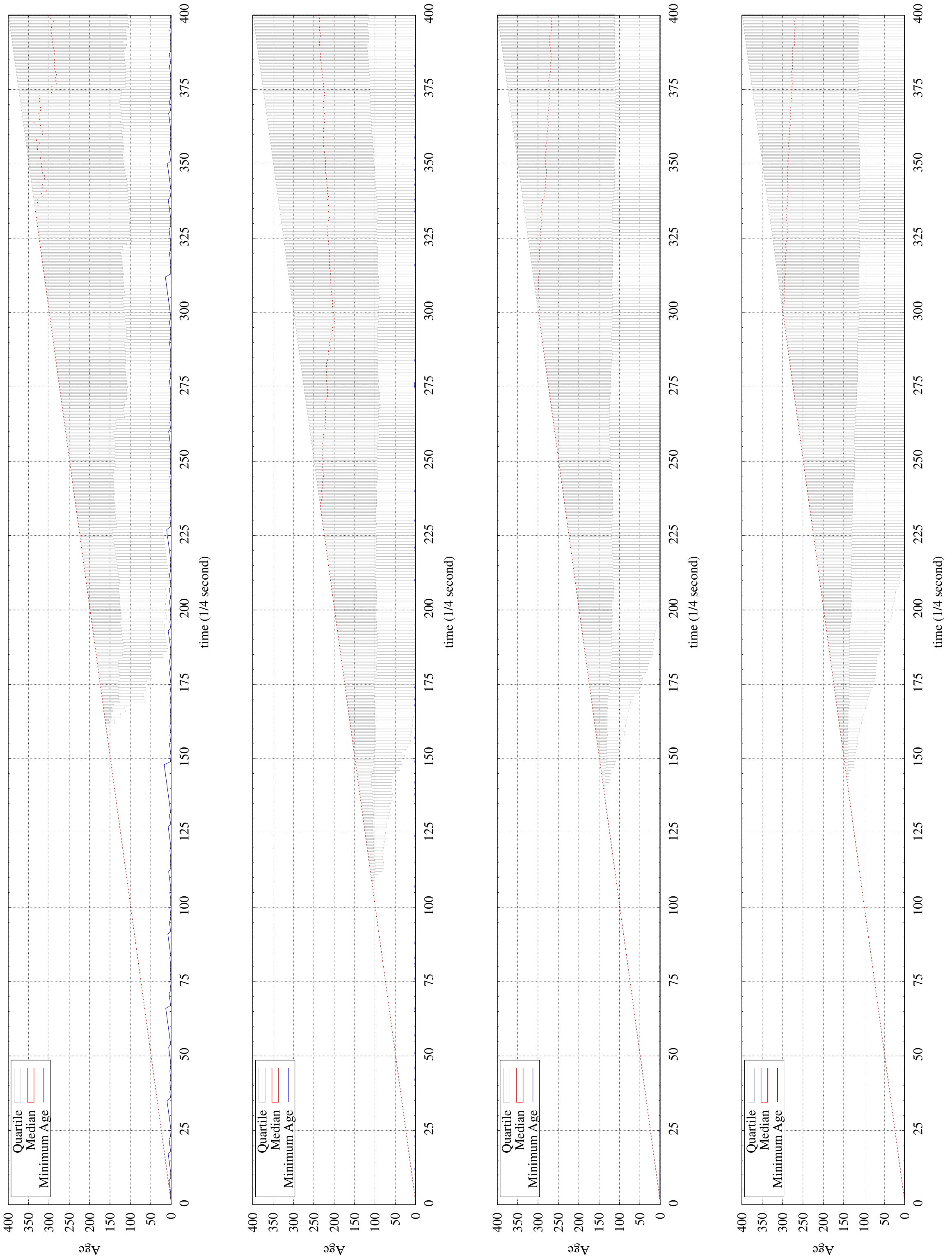


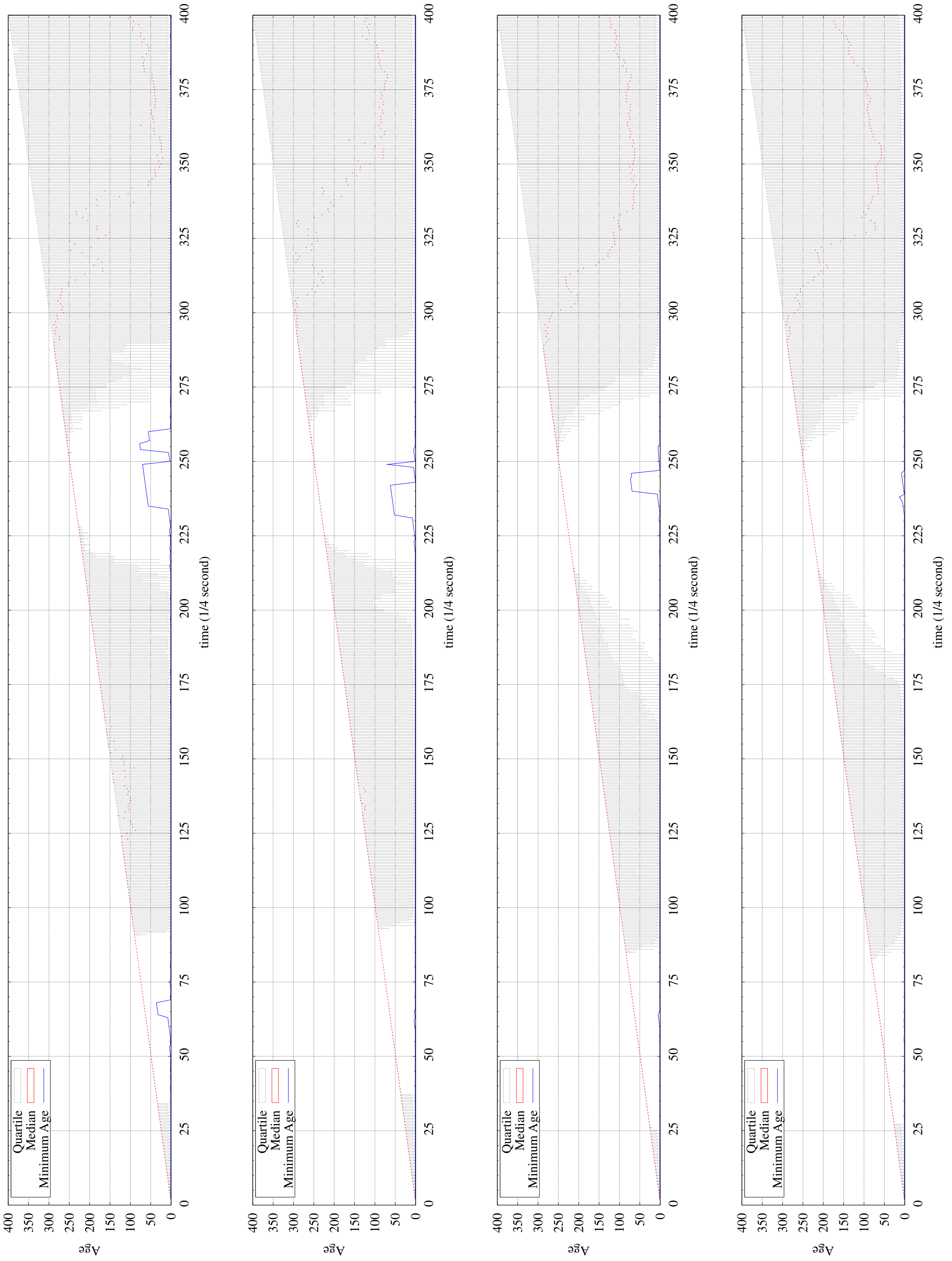
Figure 5.4: An example of box plot

50 time steps (approximately between 235 and 250 time steps of configuration 1). Hence, at this particular duration there is no new edge appearing in the network. These box plot of ages information may be used later in this manuscript when related analysis is required.



(a) Configuration 1 (b) Configuration 2 (c) Configuration 3 (d) Configuration 4

Figure 5.5: Box plot showing edge's age characteristic of each configurations of Shopping Mall Mobility Model according to Table 5.1



(a) Configuration 1 (b) Configuration 2 (c) Configuration 3 (d) Configuration 4

Figure 5.6: Box plot showing edge's age characteristic of each configurations of Highway Mobility Model according to Table 5.2

5.1.3 Assumptions

Apart from the fact that this work introduced communication protocol to realize DA-GRS reference algorithm as presented in section 4.2.3, we also specified some other assumptions here. The following section describes the datagram used in this work. While the later sub-section addresses details of the communication boundary used in GraphStream simulator.

5.1.3.1 datagram used in this work

In this work, we assume the used of Internet Protocol version 4 (IPv4) as the IP protocol. This is solely because of its popularity. With this IPv4 usage, we inherit the address space at 32 bit and the minimum-length datagram at 20 bytes (160 bits of header and without data). This Section gives the description about all communication data and its datagram size which were used in this work.

According to the realization of DA-GRS section 4.2.3, we can summarize that they are 5 message types used in this work. They are (1) *beaconing*, (2) *findingTk*, (3) *ACK_finding*, (4) *SYN/ACK_finding*, and (5) *moveTk*. Furthermore, the BREAK and CHANCE heuristic use 'break' and 'ACK_break' messages. Thus, there are altogether 7 messages types used in the proposed framework. In order to communicate with other node, we use 3 bits inside the data area of IPv4 representing the message types. Furthermore, 5 bits of data are used for giving token identification (if needs) and 32 bits of data are projection to be used in case of using TABUmove as the token traversal heuristic (Thus, it is also an option). The information of node quality using in this study are also included in beaconing message. Each quality has been given 8 bits for storing data.

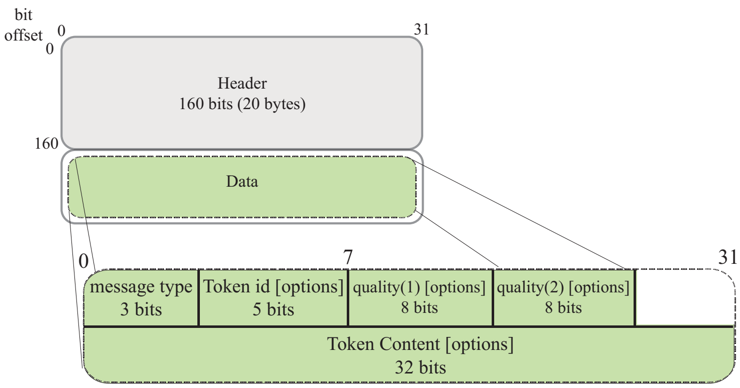


Figure 5.7: Packet Structure used for G-NODE and G-PATH heuristics

Figure 5.7 illustrates an IP packet structure based on IPv4 standard and the number of bit offset that are used for DA-GRS purpose as stated above. From this bit setting for each information we used in the algorithm, we can summarize the size of each type of packet used in this work. Table 5.3 shows the information on this size of each packet type.

5.1.3.2 Communication Assumptions in GraphStream

GraphStream is selected to replay the realistic trace files from Madhoc in a deterministic way. However, the shortcoming of GraphStream is the absence of the communication renderer.

Table 5.3: Size of each packet type used in this work

Message Type	DA-GRS Size of Packet	G-NODE/G-PATH/G-Node-Path Size of Packet
beaconing	163 bits	179 bits
findingTk	168 bits	184 bits
ACK_finding	168 bits	168 bits
SYN/ACK_finding	168 bits	168 bits
moveTk	168 bits	168 bits
moveTk using TABU{1} heuristic	200 bits	200 bits
break	163 bits	163 bits
ACK_break	163 bits	163 bits

In order to overcome this shortcoming, we limit the amount of communication by using a boundary for each nodes and for the whole simulation graph per simulation step. This section describes the detail of those communication boundaries used in this study.

Gupta et al. [138] summarizes that when n identical randomly located nodes, each capable of transmitting at W bits per second and using a fixed range, form a wireless network, the throughput $\lambda(n)$ obtainable by each node for a randomly chosen destination is:

$$\Theta\left(\frac{W}{\sqrt{n\ log\ n}}\right) bits/sec \tag{5.1}$$

This is under a noninterference assumption. Fundamentally, it is the sharing channel/resources between every node in all over the domain. Sharing resources with nodes in its local neighborhood is the reason for the constriction in capacity. According to Gupta's finding, we render our simulation by limiting throughput of each node as well as of each communication graph. According to the wireless network capacity in the Equation 5.1 above, W equals to the maximum raw data rate of used communication technology, which in this study is IEEE802.11b. Thus, W is 11 Mbit/sec. The number of n is different according to the level of limitation, at neighborhood level and at communication graph level. The number of n at the communication graph level equals to the number of all nodes in V of $G = (V, E)$. The number of n at neighborhood level of a particular node equals to the number of its neighbors. Hence, we can calculate the boundary of transmitting data of a node as:

$$\left(\frac{11Mbit/s}{\sqrt{n\ log\ n}}\right) bits/sec, \tag{5.2}$$

where n is the number of neighbor of this node.

This calculation gives us the upper bound throughput. In our simulation, we test using 50%, 75% and 100% of this maximum throughput. We found that the results given by these different setting have similar characteristics. That is the DA-GRS reference algorithm gives the lower bound both in terms of robustness and efficiency. However, the lower the throughput limitation, the lower the value of robustness metrics for both the DA-GRS reference algorithm and our proposal algorithms. We select to use the limitation at 75% of the maximum throughput because it gives a reasonable trade-off between the maximum throughput and the half-theoretical throughput. Table 5.4 shows the upper bound throughput per network with different network sizes of this study.

Table 5.4: Calculation of capacity of wireless network following Gupta's finding

Network size	Max. Throughput per Network	Max. No. of Packet (184 bits/packet)	Throughput limitation at 75%
80	934,798 bits/sec	5,080 packet/sec	3,810 packet/sec
160	614,207 bits/sec	3,338 packet/sec	2,503 packet/sec
240	482,591 bits/sec	2,622 packet/sec	1,966 packet/sec
320	407,380 bits/sec	2,214 packet/sec	1,660 packet/sec

Not only the throughput that has been limited, based on Beaconing Rate of IEEE802.11 [103], the time interval used for periodically sending the beacon is 100 millisecond. This means the beacon will be delivered 10 times per second. For our used networks, timing-snapshot is 1/4 seconds between two consecutive times and, thus, 2.5 beacons should be sent during this 1/4 seconds for one node. The limitation of beaconing also limits the number of packets that can be sent using our algorithms and heuristics.

Algorithm 6 Driving mechanism from a graph $G(t)$ to a consecutive graph $G(t+1)$ in GraphStream

t is time step in a dynamic graph $G(t)$, $t \in \{t_1, ..., t_i\}$
n_j is a node in $G(t)$, $j \in \{1, ..., j\}$
α is the maximum beaconing packet per a snapshot network $G(t_i)$
β_{n_j} is the allowed number of packet of node n_j
while $G(t) \leq G(t_i)$ **do**
  reset α and β_{n_j} for this snapshot graph
  for all n_j **do**
    if $\alpha > 0$ *and* $\beta_{n_j} > 0$ **then**
      do TM algorithms and heuristics (G-NODE, G-PATH, G-Node-Path, BREAK, CHANCE, DA-GRS)
      if n_j sends beacon **then**
        $\alpha \leftarrow \alpha - 1$
        $\beta_{n_j} \leftarrow \beta - 1$
      else
        if n_j responses to some requests and $\beta_{n_j} \geq 0$ **then**
          $\beta_{n_j} \leftarrow \beta_{n_j} - k$, k is number of packet sent
        end if
      end if
    end if
  end for
end while

5.2 Results and Analysis of the Proposal

5.2.1 Tree Traversal Strategies

5.2.1.1 Preliminary

In this section we present the comparison results obtained for the three different strategies studied for circulating the token in a decentralized tree based algorithm. These three strategies are: RANDOMmove, TABUmove, and DFSmove. The details of proposal are stated in Section 4.2.4. The first focus is on the improvement DA-GRS algorithm by our proposal tree traversal strategies. The comparison of efficiency metrics, the convergence speed of the tree and the performance ratio explained both in section 4.2.2, are presented in the following sub-section. The second focus is on the improvement of robustness metric when utilizes our tree traversal strategies on, first, DA-GRS, and later, G-NODE. The results shown are the average of simulating 100 topologies for 100 seconds each topology. For the result shown in the following sub-section, we use the set of parameters shown in Table 5.5

Table 5.5: Parameters used in the experiments

	Shopping Mall	High way
Surface (km^2)	0.32	2.24
Node density (per km^2)	347.85	72.55
Number of nodes	110	160
Avg. Number of partitions	1.95	15.90
Number of connections	446	498
Average degrees	8.13	6.23
Velocity of nodes (m/s)	0.3-3	20-40
Radio transmission range	40-80 m	
Network technology	IEEE 802.11b	

5.2.1.2 Tree Traversal Strategies over DA-GRS reference algorithm

Figure 5.2.1.2 and 5.9 show the simulation results for the shopping mall and highway environment, respectively. From both figures, DFSmove clearly gives the best behavior among these three strategies for both environments. Furthermore, both figures show the impact of the mobility model toward the resulting tree. Easily observing from Figure 5.2.1.2 (shopping mall scenario), the difference between DFSmove and the other strategies is very large. In contrary, the resulting gap from DFSmove to other strategies in the highway scenario, Figure 5.9, is not as big as what we can see in shopping mall environment. This is because of the speed of the devices and hence, the highly fluctuant topology. Thus, we measure the differences between DFSmove and the other two strategies (averaging results over the simulation time). The results of this measurement are shown in Table 5.2.1.2 to demonstrate the difference in terms of the percentile of the distance. Furthermore, as the result values do not follow a

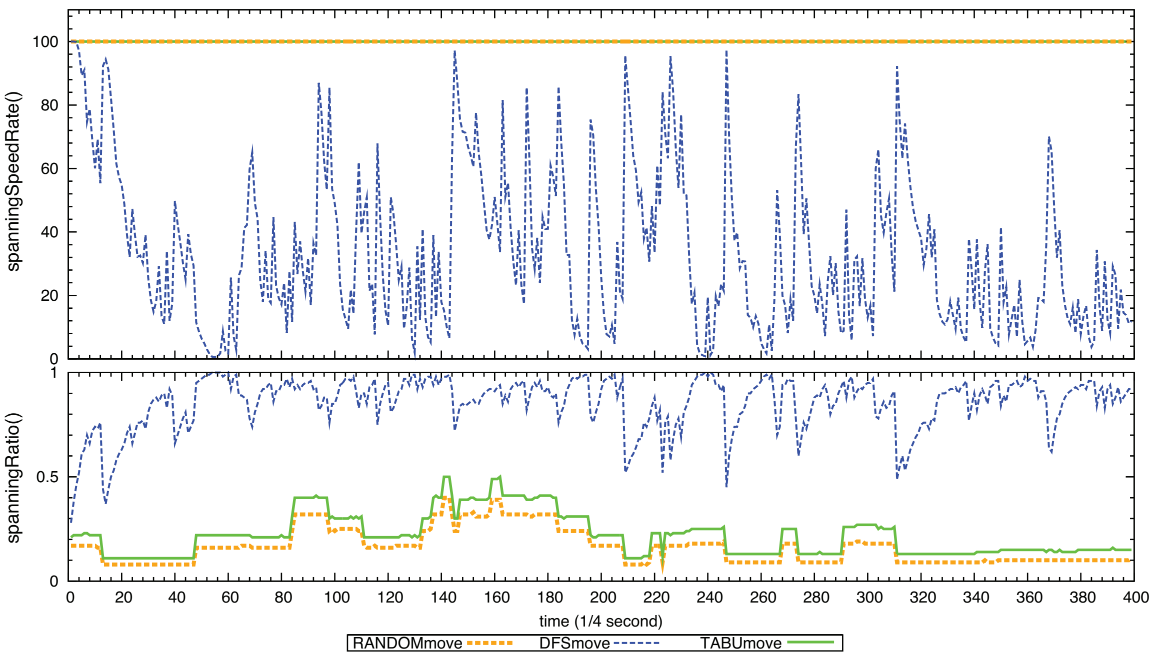


Figure 5.8: Comparison of spanningSpeedRate() measuring among all studied algorithms in 'Shopping Mall' mobility model

normal distribution in any case, we apply the Kruskal-Wallis test in order to obtain statistical significance with 95% probability in our comparisons. The results show that DFSmove are significantly better than TABUmove and RANDOMmove.

Table 5.6: The percentile of the distance from DFSmove to the other strategies

		TABUmove	RANDOMmove
Highway	spanningRatio	23.71%	34.71%
	convergenceSpeed	12.38%	12.40%
Mall	spanningRatio	63.70%	69.19%
	convergenceSpeed	66.42%	66.42%

According to Table 5.2.1.2, the differences are up to 60-70% when compare the result of DFSmove and the other strategies in shopping mall model. On the other hand, for highway model, the differences of results between all the strategies are distinguishable and also statistically significant, but not so huge different (12-35% of differences) as found in shopping mall model. This comes from the fact that the highway model has a high fluctuating mobility. Thus, the topology is more likely to change than in the shopping mall.

The overall results show that the RANDOMmove strategy is the worst one in both environments. This behavior was expected, since when using the random technique many nodes in the tree can hardly possess the token, so the merging trees in those areas rarely happened. TABUmove ensures that one neighbor will not possess the token twice consecutively. Thus, TABUmove achieves better distribution of the token than RANDOMmove.

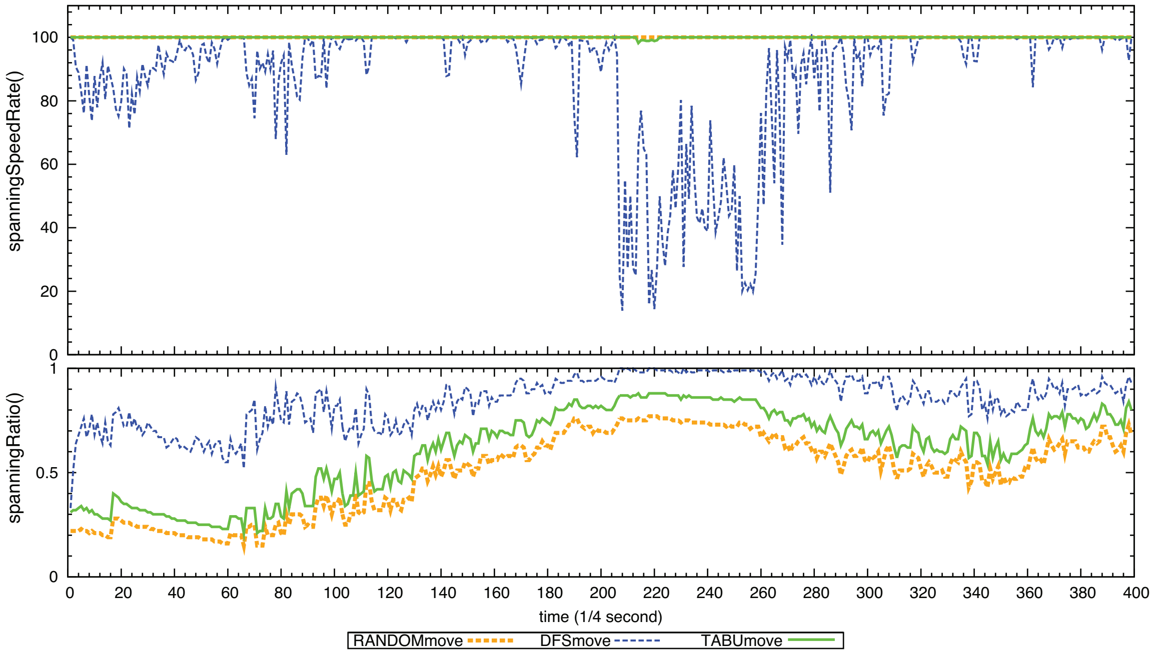


Figure 5.9: Comparison of spanningSpeedRate() measuring among all studied algorithms in 'highway' mobility models

As stated at the beginning of this paper, our intuition suggest that a strict ordering strategy may not be a promising technique for a high changeable topology. The reason to this intuition is that the topology is changing a lot in a short time while the token is moving mostly in the same area during such small period of time. However, the experimentation results deny our intuition. Our results show that DFSmove behaves better than RANDOMmove and TABUmove. Thus, it can be confirmed that an ordering strategy like DFSmove can work well under highly changing topology.

Table 5.7: Multi-objective multi-constraint study

	DFSmove	TABUmove	RANDOMmove
Memory required	Yes	Yes	No
Token memory	No	Yes	No
Node memory	Yes	No	No

In Table 5.7 we resume the behavior of each technique since we can consider this is a multi-objective multi-constraint problem, and depending on the necessities of each situation a technique or another can be used. As it can be seen in Table 5.7 the only strategy that uses no memory at all is RANDOMmove. For those using memory it is possible to distinguish between using the memory in the node like DFSmove, or using memory in the token as TABUmove.

In order to confirm the benefit of DFSmove, we also give an experimentation and result of comparing DFSmove, TABUmove, and RANDOMmove strategies used DA-GRS algorithm. The experimentation is simple. The number of iterations used for merging spanning trees has been measured. We utilized the same graph and experimentation method that have been used to confirm the benefit of TABUmove found in [91]. The used graph contains a connected component of 170 nodes and has been produced from random waypoint mobility model. The result shown in Figure 5.10 is an average result from 600 runs of 1000 iterations (but illustrated here only the first 200 steps since DFSmove converges at 154 iterations and there is no further change in the near range).

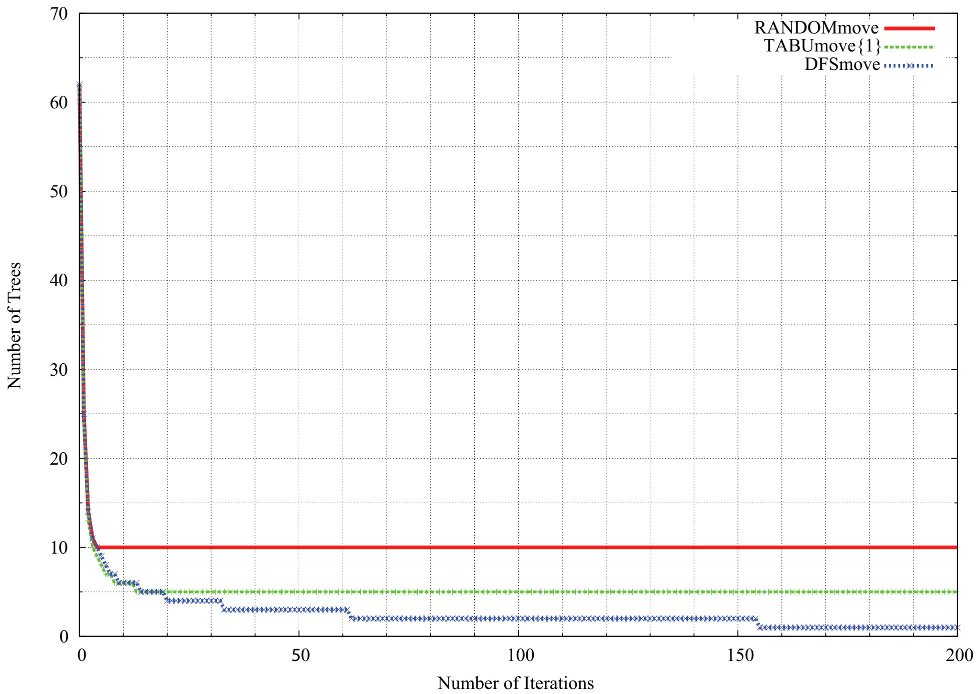


Figure 5.10: The result based on 600 runs comparing the number of iteration used in obtaining trees using different strategies (RANDOMmove, TABUmove and DFSmove)

From the result, DFSmove performs the best over time. The more number of iterations, the less the number of trees. We can observe that around 154 iterations, all nodes converge into the same tree.

To conclude, the DFSmove heuristic performs merging tree better than TABUmove and RANDOMmove over time and finally can produce a tree over a connected component in shorter time.

5.2.1.3 Tree Traversal Strategies over the Robustness of Trusted Tree Topology by G-NODE

With closely studying the mechanism of G-NODE 4.6.2.1, we foresee that the token behavior has also an impact on the robustness of the spanning forest. These motivate us (1) to investigate a new token traversal in high mobility network using G-NODE algorithm, and, (2) to determine a suitable strategy, such that the weighted spanning trees (created by G-

NODE) are robust to the dynamic features of the underlying DT-MANETs [ICUMT09]. Hence, the objectives of the work in this section are focusing on the enhancement of the robustness metrics. The results of efficiency metrics have been shown along side with the robustness metrics in Figure 5.11 and 5.12.

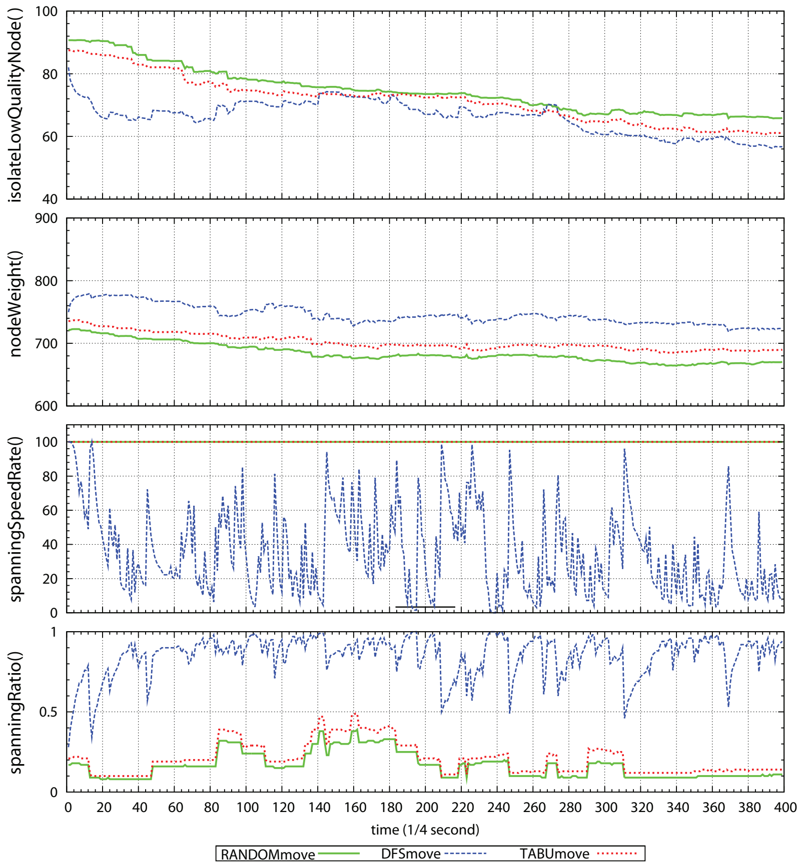


Figure 5.11: Comparison of spanningRatio(), spanningSpeedRate(), nodeWeight() and isolatingLowQualityNode() measuring among all studied algorithms in Shopping Mall Mobility Model

Since, the ideal *spanningRatio*() is at one, which means there exists only one tree within one connected component, both Figures 5.11 and 5.12 show the best performance of DFSmove strategy. Indeed, among the three strategies, DFSmove achieved significantly higher quality in all aspects except for isolatingLowQualityNode() in both mobility models.

From the results, DFSmove can give a lower number of spanning trees than the other heuristics. This means DFSmove creates a bigger trusted spanning tree and thus *nodeWeight*() value can be expected to be higher than several smaller trusted spanning trees. While the other two cannot merge or collect spanning trees efficiently, there is more chance for low-trusted nodes on each small spanning tree to be at leaves because of the G-NODE algorithm itself. Hence, the cost function on *isolatingLowQualityNode*() can give a good result when there are more spanning trees in a forest, which is contrast to the objective of *spanningRatio*() and this study. This fact makes it possible to have value of isolatingLowQualityNode() from

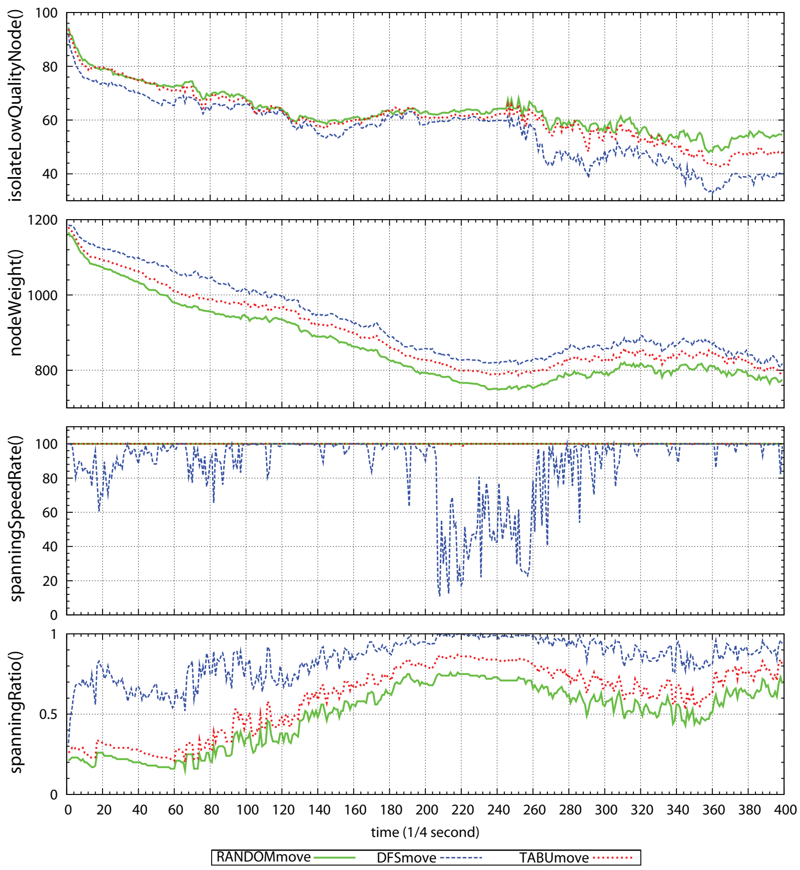


Figure 5.12: Comparison of spanningRatio(), spanningSpeedRate(), nodeWeight() and isolatingLowQualityNode() measuring among all studied algorithms in Highway Mobility Model

G-NODE with DFSmove lower than DA-GRS with randomness alone, if the low quality nodes are in articulation position.

Figure 5.13 shows this contrasting fact. On sub-figure 5.13(a), there is only one tree spans on a connected component while it is not the case on the Sub-Figure 5.13(b). In Figure 5.13(a), node y has the quality value equals to threshold and it is an articulation point. In this case, the isolateLowQualityNode() will suffer from the fact that low quality node is in articulation point. In Figure 5.13(b), the benefit of G-NODE heuristic makes those low quality node to be pushed to the leave position of tree when possible. The resulting found in this Sub-Figure 5.13(b) is that isolatingLowQualityNode() function yielding very high robustness; given threshold of being low quality node equals to 0.2, this sub-figure succeeded in giving 100% of isolating low quality node.

To summarize, DFSmove achieved the best *spanningRatio*() among all three algorithms in both mobility models. This means using DFSmove heuristic as a way to conduct the movement of token is the most efficient way for merging all nodes in the same connected component to be in the same tree. *spanningSpeedRate*() shows how fast all three strategies can influence the convergence of a tree on a connected component of a network in both mobility models. From both figures, it is apparent that DFSmove uses lower rate compare

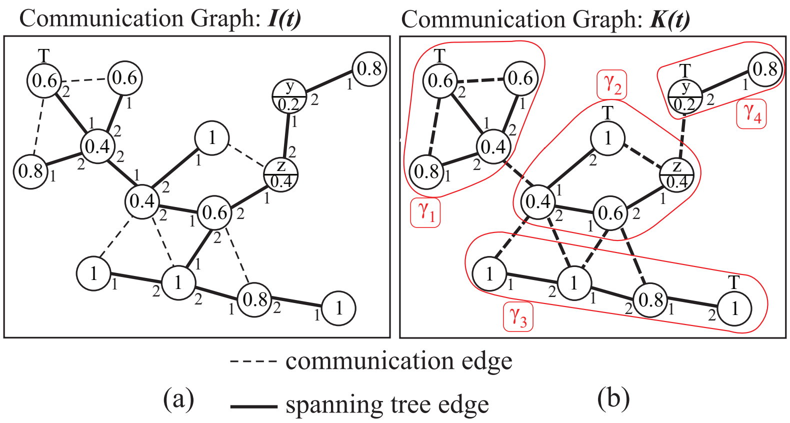


Figure 5.13: Example of Contrasting between spanningRatio() and isolatingLowQualityNode()

to other heuristics. Hence, it can be concluded that DFSmove can create trusted spanning forest faster than other strategies.

Conclusion can be summarized into each point of interest as follow. (1) Two different approaches on information storage are presented (token's memory: TABUmove, nodes' memory: DFSmove). (2) DFSmove yields better results in terms of robustness, performance and convergence speed. (3) DFSmove might not be applicable with very small memory nodes in very dense network. (4) From the results, the trade-off between having a high spanningRatio() and having many highly robust trees but formed into smaller size has been raised. This point also leads to the decision to drop the cost function *isolatingLowQualityNode*() later, since it gives a contrast meaning to the objective of *spanningRatio*() and this study.

5.2.2 Robust topology utilizing Cooperative Enforcement Approach

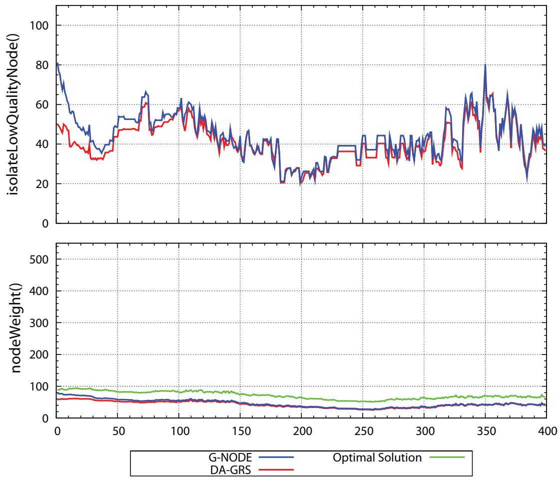
The experimentation under this section are done according to the proposal in section 'Robust Spanning Tree by Nodes' property: Cooperative Enforcement Approach' as described in 4.3.1. Accordingly, the result and analysis belong to a single objective which is to enhance the robustness of spanning forest topology by utilizing trust value. Please note that this work only utilizes such trust value literally coming from cooperative enforcement paradigm. Therefore trust value is assumed and assessing the trust value is out of the scope of this work.

The algorithm and heuristics used for this work are G-NODE using DFSmove as the token traversal strategy. The weight of each node, n, is trust level, denoted by $trust(n)$, where $0 < trust(n) \leq 1$. Given $\Theta_t = \{n' \in V_t(G) | trust(n') \leq threshold\}$ be the set of all low-trustable nodes at moment t. The threshold is 0.2. The validation of result is done by comparing the robustness metrics of results from G-NODE, DA-GRS reference algorithm and optimal value over simulation time. The results shown are the average of simulating 100 topologies for 100 seconds in each topology. The result are discussed in the following sub-section, we use the set of parameters shown in Table 5.1 and Table 5.2 for shopping mall and highway mobility model respectively. We performs the comparison on both robustness and efficiency metrics. Here presents the main comparison which are the robustness metrics. The full comparison can be found in Appendix A.1.

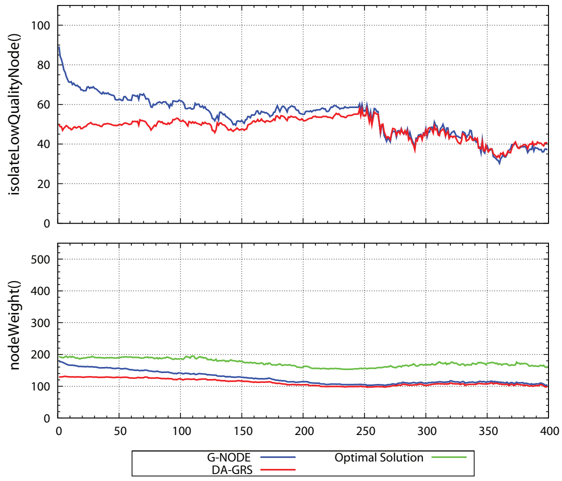
Figure 5.14 comprises of four sub-figures showing the result comparison between DA-GRS reference algorithm as an lower-bound and our G-NODE equipped with DFSmove heuristic. The optimal value is shown for nodeWeight() function. Each figure presents the result from different configuration of network, which has been introduced in Table 5.1 and 5.2. As can be seen, the value of nodeWeight() function grows from configuration 1 to 4 since the number of node and edges grow.

The effect of DFSmove heuristic equipped with G-NODE cannot be seen in these set of results. As discussed earlier, DFSmove achieves the faster rate in order to span tree. Thus, G-NODE tends to give a bigger tree than DA-GRS reference algorithm. Accordingly, it is harder to achieve higher quality for the isolateLowQualityNode() function, if such low quality nodes are in articulate position. Unlike the results from previous Section 5.2.1.3, the networks used in this particular work does not seem to have problem with articulation nodes. In all sub-figures, G-NODE gives higher quality of the isolateLowQualityNode() function than DA-GRS reference algorithm. Furthermore, G-NODE gives higher quality of nodeWeight() function than DA-GRS reference algorithm in all experimentation also.

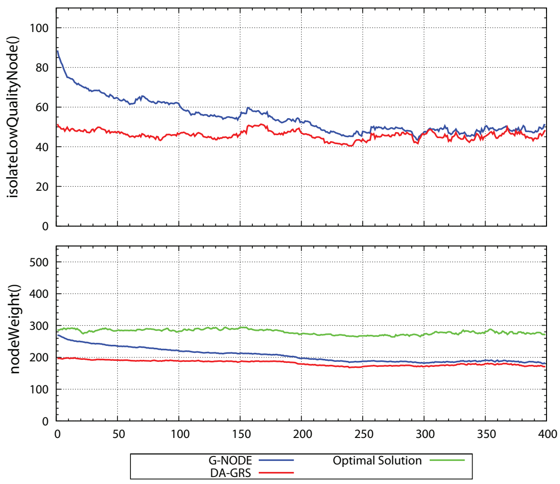
G-NODE gives results with very high distance from results of DA-GRS reference algorithm at the beginning of the simulation time and then declining down but yet higher than DA-GRS reference algorithm. Actually, this behavior is expected when we proposed G-NODE. G-NODE operation aims at merging process, but it has no mechanism to relocate the existing tree in order to enhance the robustness quality. This shortcoming leads to the introduction of BREAK and CHANCE heuristics which is discussed later in Section 5.2.5 and 5.2.6 respectively. Another interesting point is at the very beginning of the simulation that the ability of G-NODE, a local decision heuristic, can almost reach the optimal value. However, with the lack of ability to improve the spanning forest over time, the robustness of spanning forest deteriorates.



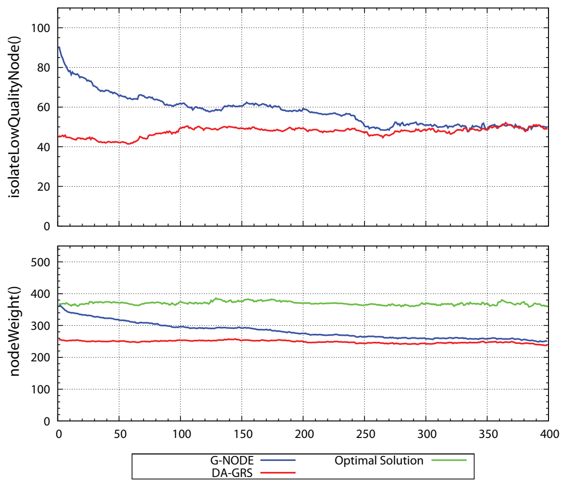
(a) Result from Highway Configuration 1



(b) Result from Highway Configuration 2

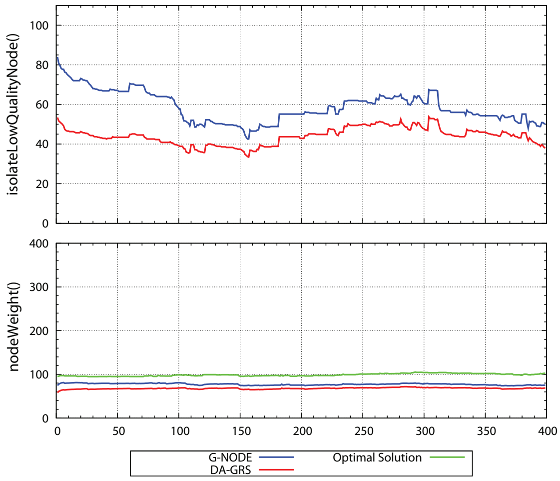


(c) Result from Highway Configuration 3

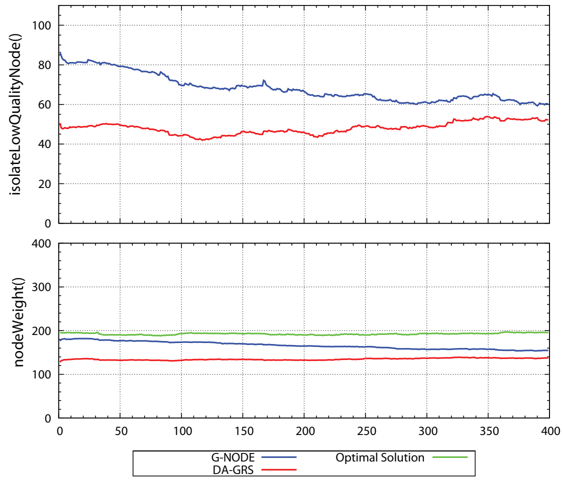


(d) Result from Highway Configuration 4

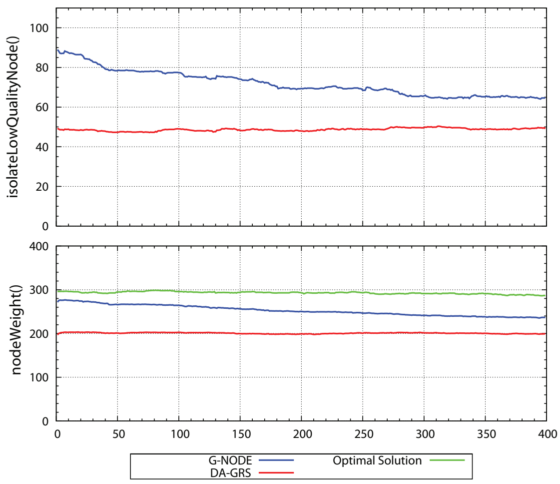
Figure 5.14: Result of G-NODE utilizing trust value compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model



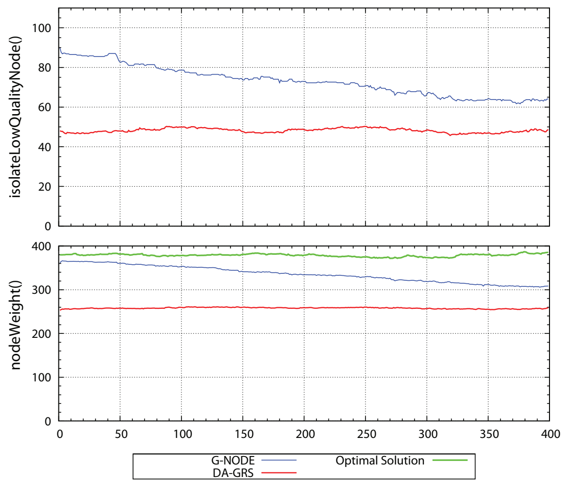
(a) Result from Mall Configuration 1



(b) Result from Mall Configuration 2



(c) Result from Mall Configuration 3



(d) Result from Mall Configuration 4

Figure 5.15: Result of G-NODE utilizing trust value compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model

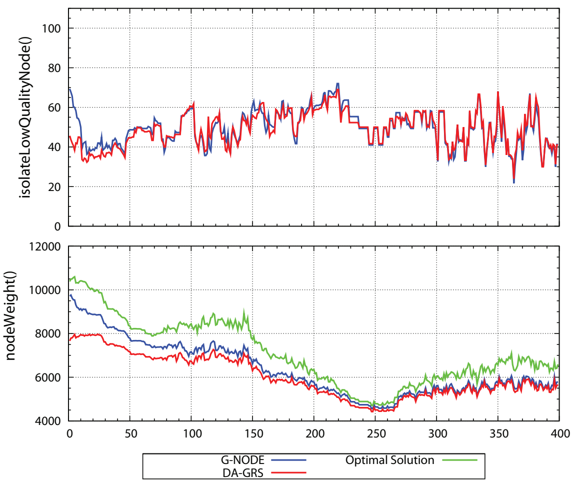
Conclusion can be summarized into each point of interest as follow. (1) The enhancement of robustness quality of spanning forest topology can be done using only one-hop information. (2) Greedy heuristic is used in G-NODE algorithm selecting better quality nodes. This is done de-centrally and improves the effectiveness of the overall topology substantially. (3) The lower the mobility of nodes, the easier in obtaining the robustness spanning forest. This can be notice from the differences between results of two different mobility models, highway and shopping mall.

5.2.3 Robust topology utilizing Energy Availability Approach

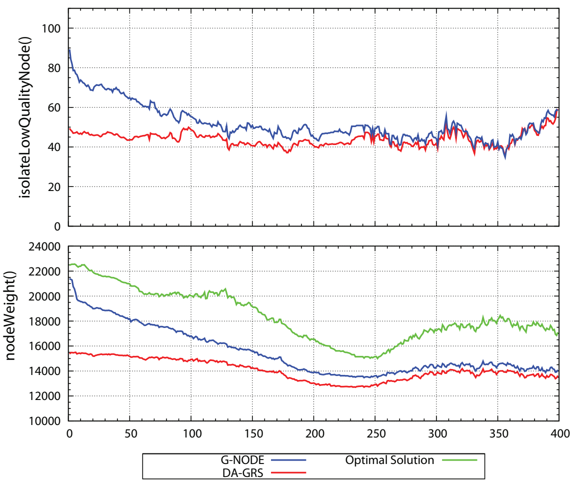
The experimentation under this section are done according to the proposal as described in Section 4.3.2, 'Robust Spanning Tree by Node's Property: Energy Availability Approach'. The result and analysis belong to a single objective which is to enhance the robustness of spanning forest topology by utilizing battery level. The algorithm and heuristics used for this work are G-NODE using DFSmove as the token traversal strategy. The weight of each node, n, is battery percentage that is left on each node n, denoted by $battery(n)$, where $0 < battery(n) \leq 100$. Given $\Theta_t = \{n' \in V_t(G) \mid battery(n') \leq threshold\}$ be the set of all low-battery nodes at moment t. The threshold is 20. The validation of result is done by comparing the results of robustness metrics from G-NODE, DA-GRS and optimal value over simulation time. The results shown are the average of simulating 100 topologies for 100 seconds in each topology. The result shown in the following sub-section, we use the set of parameters shown in Table 5.1 and Table 5.2 for shopping mall and highway mobility model respectively. Please note that the comparison on both robustness and efficiency metrics takes a lot of space. Thus, we only present the main comparison which are the robustness metrics here. However, the full comparison can be found in Appendix A.2.

Figure 5.16 and 5.17 confirm the benefit of G-NODE heuristic for enhancing robustness of spanning forest over DA-GRS. Although, the nodeWeight() value of G-NODE cannot achieve the optimal solution, we see the very close range to optimality at the very beginning of the simulation. This is because of the fact that G-NODE constructs the spanning forest from initial stationary state. However, it still lacks of changes management to improve the existing spanning forest.

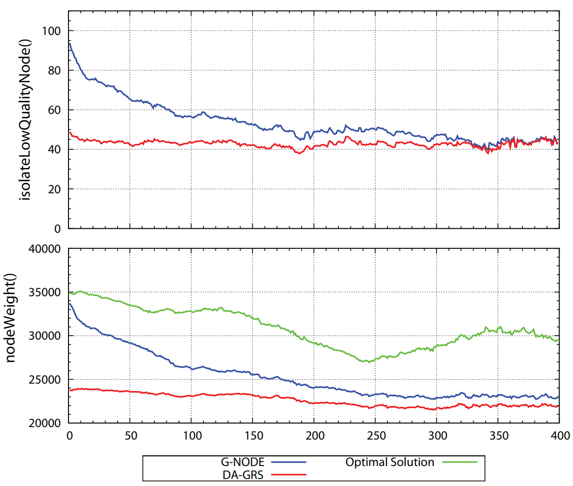
Better result was obtained on isolateLowQualityNode() value comparing G-NODE and DA-GRS reference algorithm. This also reveals the importance of position of low quality node. In this case, there is a high probability that low quality node is not in an articulate position. Thus, G-NODE is capable of pushing such nodes to the leave of tree. In conclusion, G-NODE is used to construct and maintain robust spanning tree topology and able to maintain robustness in terms of availability of energy level in each mobile node.



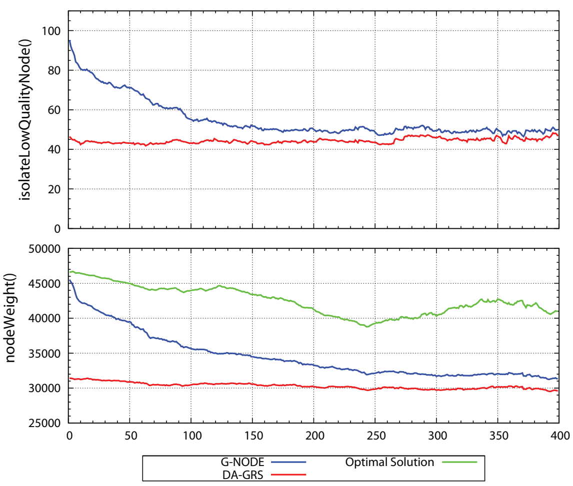
(a) G-NODE on Highway Configuration 1



(b) G-NODE on Highway Configuration 2

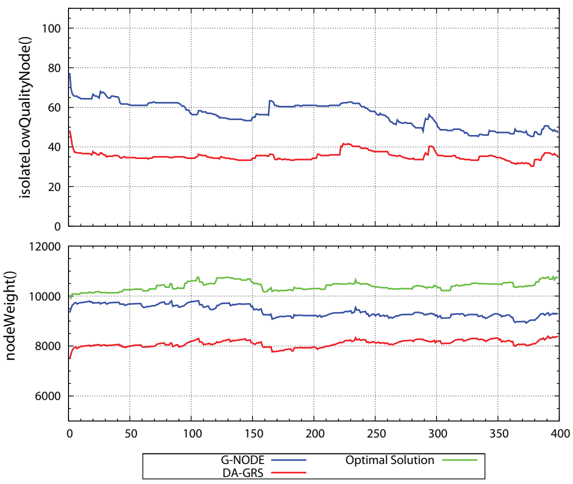


(c) G-NODE on Highway Configuration 3

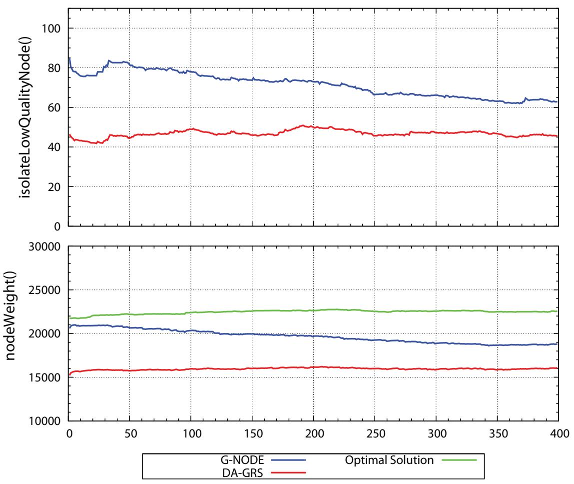


(d) G-NODE on Highway Configuration 4

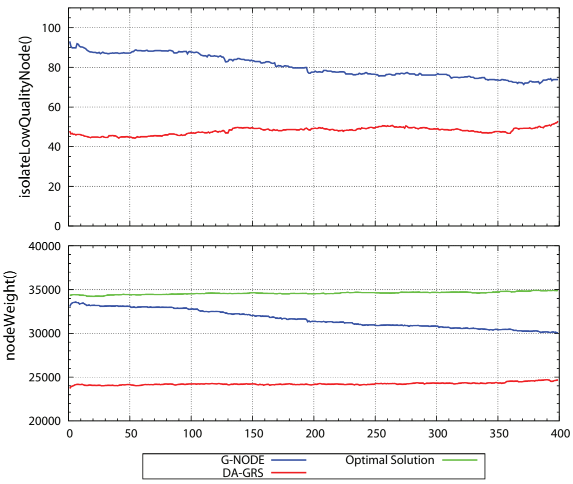
Figure 5.16: Result of G-NODE utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model



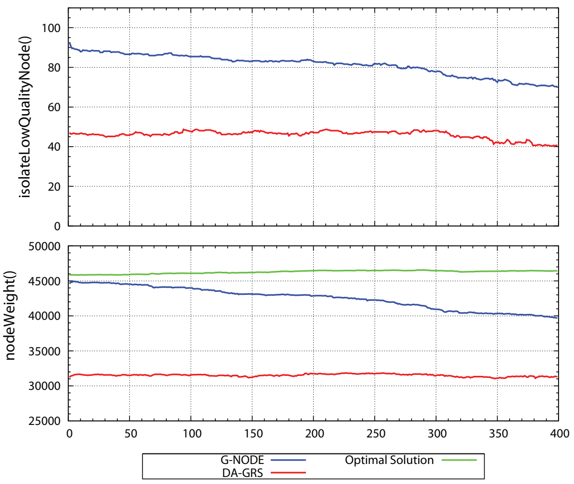
(a) G-NODE on Mall Configuration 1



(b) G-NODE on Mall Configuration 2



(c) G-NODE on Mall Configuration 3



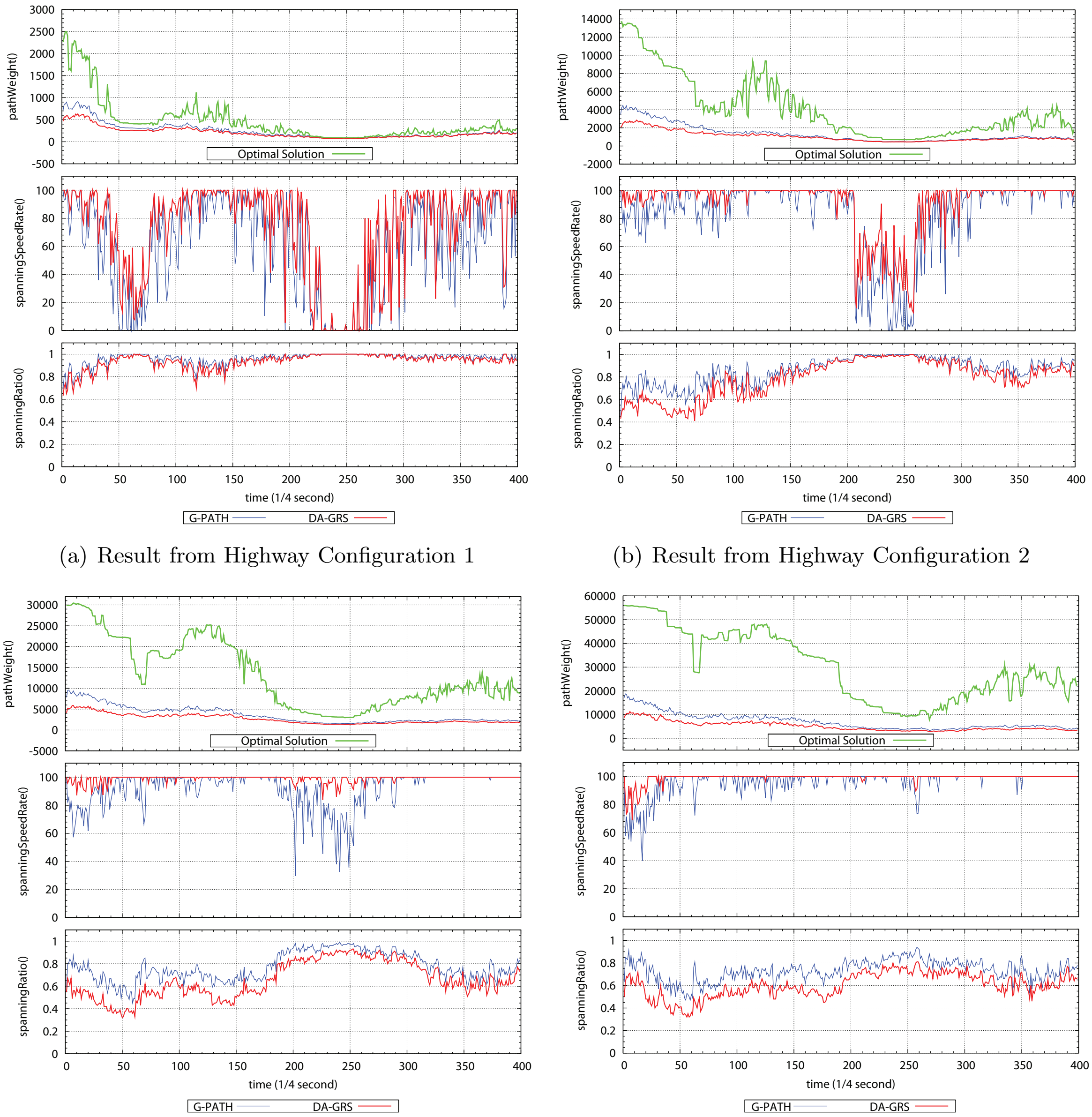
(d) G-NODE on Mall Configuration 4

Figure 5.17: Result of G-NODE utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model

5.2.4 Robust topology utilizing Capacity Bandwidth Approach

The experimentation under this section are done according to the proposal in Section 4.3.3. The result and analysis belong to a single objective which is to enhance the robustness of spanning forest topology by focusing on quality of path used in such topology. The quality is the capacity bandwidth (CB). The results shown here are the average of simulating 100 topologies for 100 seconds in each topology. Again, we use the set of parameters shown in Table 5.1 and Table 5.2 for shopping mall and highway mobility model respectively.

Results of both highway and shopping mall mobility models are shown in Figure 5.18 and 5.19. It is clearly seen that the optimal value is a lot higher when number of nodes increasing (80, 160, 240 and 320 nodes for configuration 1 to 4). Although our decentralized algorithm (G-PATH) can construct a better solution than DA-GRS reference algorithm, the robustness gap between G-PATH and optimal solution is still very high in configuration 2-3 and as shown in Figure 5.18(b), 5.18(c) and 5.18(d). Actually, this big difference of result value is not so surprising since the pathWeight() function also captures the characteristic of spanningRatio() function. That is if there are more than one trees per a connected component, the pathWeight() value will decreasing rapidly, this fact is shown in Figures 5.18 and 5.19. For example, at sub-figure 5.18(d), spanningRatio() shows us that both G-NODE and DA-GRS reference algorithm cannot span a single tree over connected component(s) of the configuration 4 graph. Hence, there exists a large gap between optimal value and G-NODE in pathWeight() function.



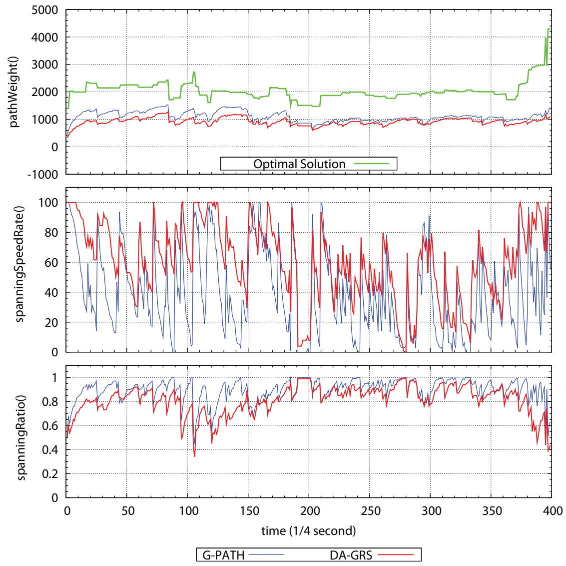
(a) Result from Highway Configuration 1

(b) Result from Highway Configuration 2

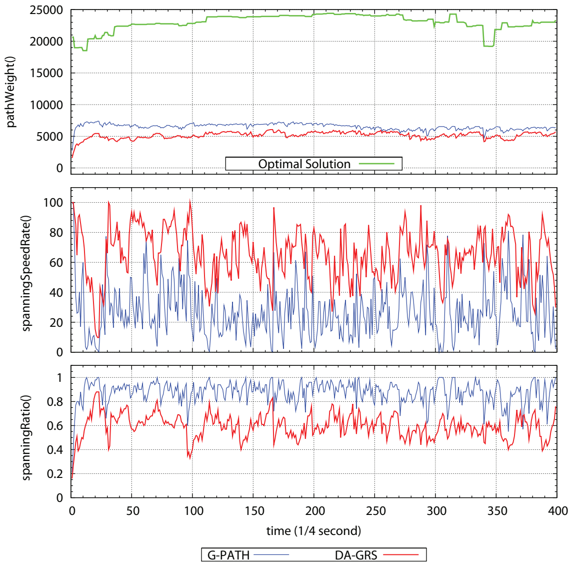
(c) Result from Highway Configuration 3

(d) Result from Highway Configuration 4

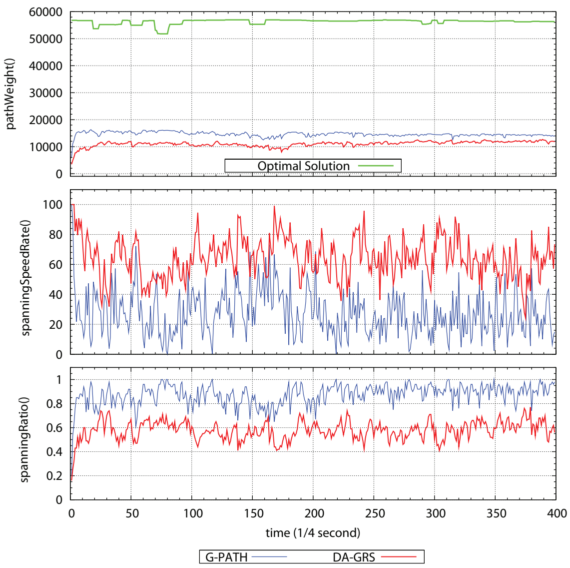
Figure 5.18: Result of G-PATH utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Highway Mobility Model



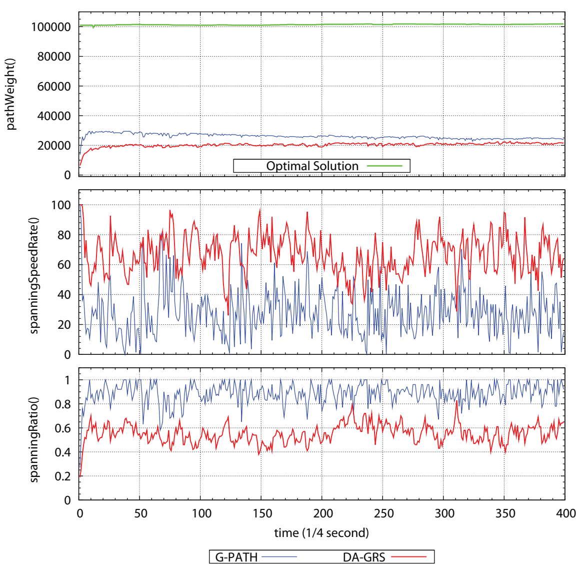
(a) Result from Mall Configuration 1



(b) Result from Mall Configuration 2



(c) Result from Mall Configuration 3



(d) Result from Mall Configuration 4

Figure 5.19: Result of G-PATH utilizing battery percentage level compare to DA-GRS and optimal value of nodeWeight() function: Mall Mobility Model

5.2.5 Reorganization of topology by BREAK heuristic

As presented in [EUC08], we performed discrete-time simulations. So the communication network corresponds to a series of static graphs: $G(t)$ for $t \in \{t_1, t_2, t_3, ..., t_{40}\}$. The experiment provided the comparison for the 3 studied algorithms under investigation in the two real-world mobility models. The algorithms are DA-GRS (as a lower bound), G-NODE, and G-NODE BREAK with different T_break at every 1.25, 2.5, and 5 seconds. The simulations were done using 100 runs per algorithm per mobility model. This work assumes 5 different levels of $trust(n)$ where $trust(n)$ equal to 1 is the lowest trust level and 5 is the highest. The *threshold* value used to determine the trustability of any node is equal to one (any node have trust level equal to one is a low-trustable node). Two real-world mobility models, 'shopping mall' and 'highway', are used in the simulations using the parameters summarized in Table 5.8

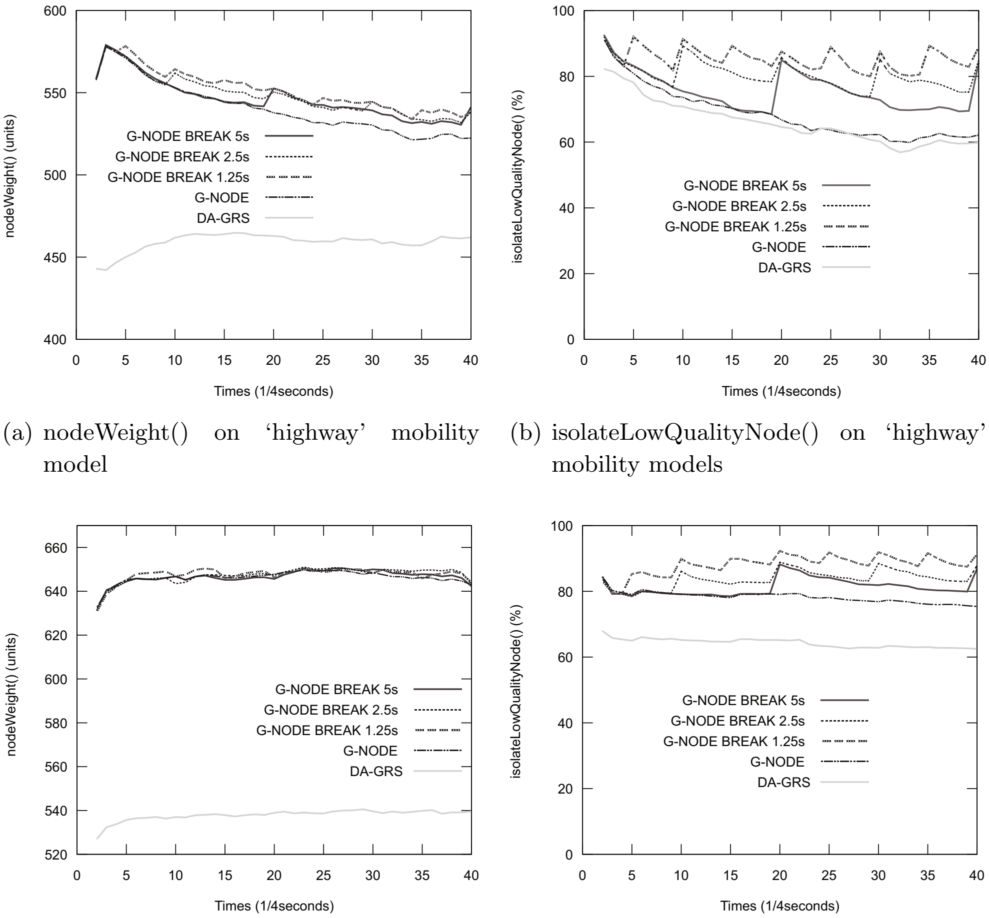
Table 5.8: Parameterization used in Madhoc

	Shopping Mall	High way
Surface (km^2)	0.32	1.00
Node Density (per km^2)	1000	80
Number of Nodes	100	80
Avg. Number of Partitions	2.68	1.70
Number of Connections	389	405
Average Degrees	7.82	10.17
Velocity of Nodes (m/s^{-1})	0.3-3	20-40
Radio Transmission Range	40-80 m	

On each graph, details of each cost function applied to all studied algorithms have been shown. From those graphs, 5.20(a) and 5.20(b), it can be seen several jumps on both cost functions when applies 'BREAK'. These high jumps confirm the benefit of 'BREAK' in improving the quality of trusted spanning tree.

G-NODE yielded a better trusted spanning tree over DA-GRS in both network models (highway and shopping mall). However, comparison among algorithms with 'BREAK' heuristic and G-NODE in shopping mall model did not reveal significant difference. This is because of the typical topology of shopping mall itself. Since nodes in this model move slowly, the changing or dynamic of the topology is very low. This implies that attempting to reorganize spanning trees may not be necessary as this opportunity rarely occurs. On the other hand, nodes in highway model have high mobility, this is when 'BREAK' heuristic can be advantageous.

In summary, G-NODE is able to select more robust (trustable) trees and the quality of these trees is superior than the random trees created by DA-GRS. However, G-NODE has no ability to reorganize the created trees in order to maintain efficient trusted spanning trees in dynamic topology of network. Hence, the BREAK heuristic is incorporated in G-NODE algorithm to overcome this shortcoming as shown in the all figures. The work also affirms the importance of the ability to adapt trusted spanning tree into a changing of environment. While this adaptation may produce more overhead in G-NODE with BREAK heuristic using



(a) nodeWeight() on 'highway' mobility model

(b) isolateLowQualityNode() on 'highway' mobility models

(c) nodeWeight() on 'shopping mall' mobility model

(d) isolateLowQualityNode() on 'shopping mall' mobility model

Figure 5.20: Comparison of nodeWeight() and isolateLowQualityNode() functions measuring among all studied algorithms

T_break at 5 seconds (5*s*) as illustrated in Tables 5.10 and 5.9. The others produced lower overhead than G-NODE alone. This is because the low-trusted node has been placed in a suitable position for the quality of tree, there is no need to change or apply the BREAK heuristic again and again. As soon as the tree becomes more robust, the overhead will be reduced. Hence, it can be said that the BREAK heuristic can both help in increasing the quality of the trusted spanning tree and reducing the overhead over time. A crucial issue in using BREAK heuristic is how to choose the suitable *T_break* to apply the heuristic in a dynamic way. In the context of this study, and the mobility model used, the comparison among result from G-NODE BREAK depicts that the best *T_break* is at every 1.25 seconds.

In conclusion, this work proposes further improved for G-NODE with the implementation of the BREAK heuristic. G-NODE is a local decision making heuristic based on greedy algorithm. Its main property is to construct spanning forest regarding to some quality value on mobile nodes. The benefit of G-NODE is high at the construction phase of topology. However, it incapable of improving the topology dynamically over time. Hence, BREAK heuristic is implemented to allow dynamic reconfiguration of such robust spanning trees topology.

Table 5.9: Overhead comparison among 'BREAK' heuristic at different *T_break* in G-NODE on 'highway' model

	G-NODE	G-NODE BREAK 1.25s	G-NODE BREAK 2.5s	G-NODE BREAK 5s
Packet Used	1978.89	2107.75	2022.58	1988.44
Low-quality Node Break	0	36.39	24.09	14.55
Created Token from Break	0	79.35	53.27	34.00

Table 5.10: Overhead comparison among 'BREAK' heuristic at different *T_break* in G-NODE on 'shopping mall' model

	G-NODE	G-NODE BREAK 1.25s	G-NODE BREAK 2.5s	G-NODE BREAK 5s
Packet Used	2743.13	2446.61	2462.04	2787.36
Low-quality Node Break	0	45.02	25.32	12.75
Created Token from Break	0	93.70	53.75	27.67

5.2.6 Reorganization of topology by CHANCE heuristic

The experiment provided the comparison for the 3 studied algorithms under investigation in two real-world mobility models. The algorithms are DA-GRS reference algorithm (as a lower bound), G-NODE, and G-NODE CHANCE. Apart from that, the optimal value of nodeWeight() function is provided. The simulations were done using 100 runs per algorithm per mobility model. The weight of each node, n, is battery percentage that is left on each node n, denoted by $battery(n)$, where $0 < battery(n) \leq 100$. Threshold is set to 20. Two real-world mobility models, 'shopping mall' and 'highway', are used in the simulations using the parameters as stated in Table 5.1 and Table 5.2 for shopping mall and highway mobility model respectively. The results shown in Figure 5.21 and 5.22 are an example of the results of configuration 4 of both mobilities. The rest of results can be found in Appendix A.3.

As we can observe from these results, G-NODE CHANCE provides the best quality spanning forest solutions in terms of robustness over both mobility models. These solutions almost reach the optimum value of nodeWeight() function for both mobility models. For the spanningSpeedRate(), when the network is bigger, it can be seen clearly that the DA-GRS reference algorithm is the slowest algorithm to span trees. Also, there are not so distinctive between G-NODE and G-NODE CHANCE from both mobilities. However, it can be observed that in highway mobility that at some duration of simulation, G-NODE is able to converge spanning forest faster than G-NODE CHANCE and the performanceRatio() of G-NODE is also better accordingly. This evidence should come from the fact that CHANCE do break the existing tree in order to re-locate some nodes and gain more robust solution trees. Spending times on those tasks has an effect to the overall efficient metrics. Nevertheless, overhead in computation time is negligible.

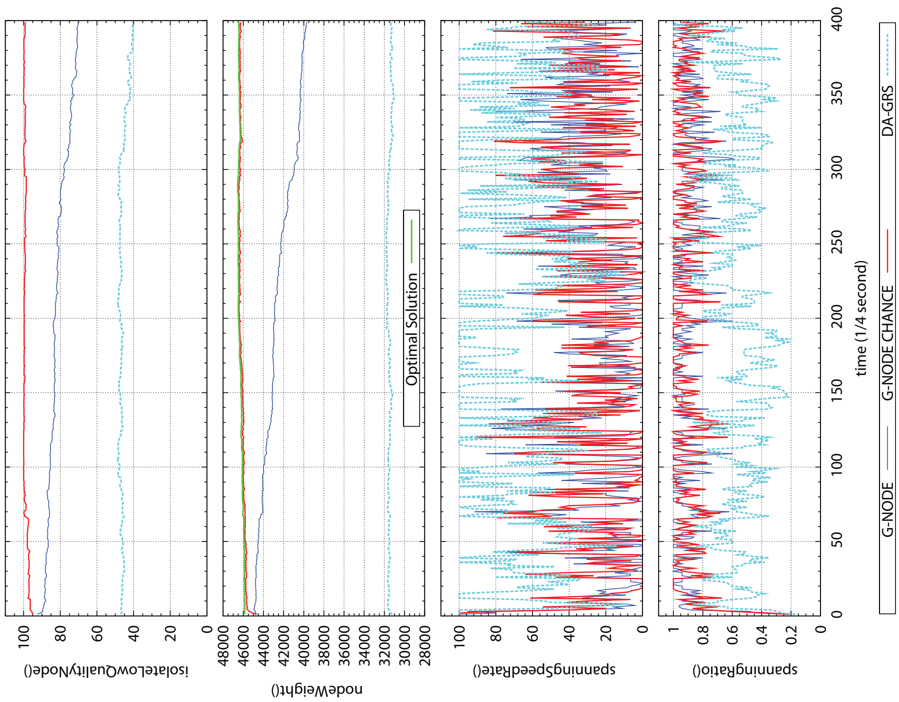


Figure 5.21: G-NODE CHANCE on Shopping Mall Configuration 4

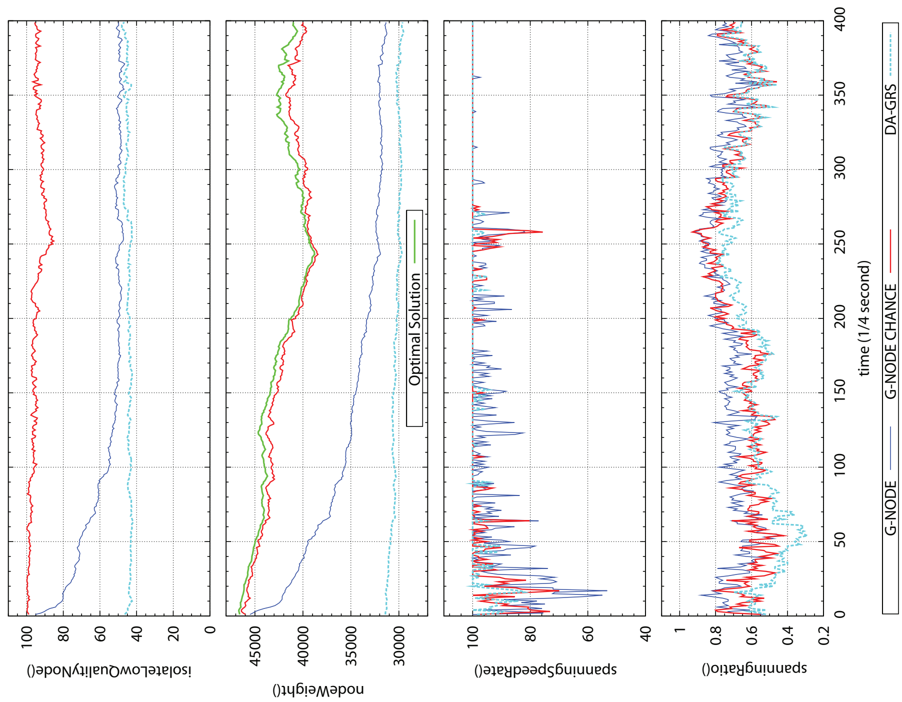
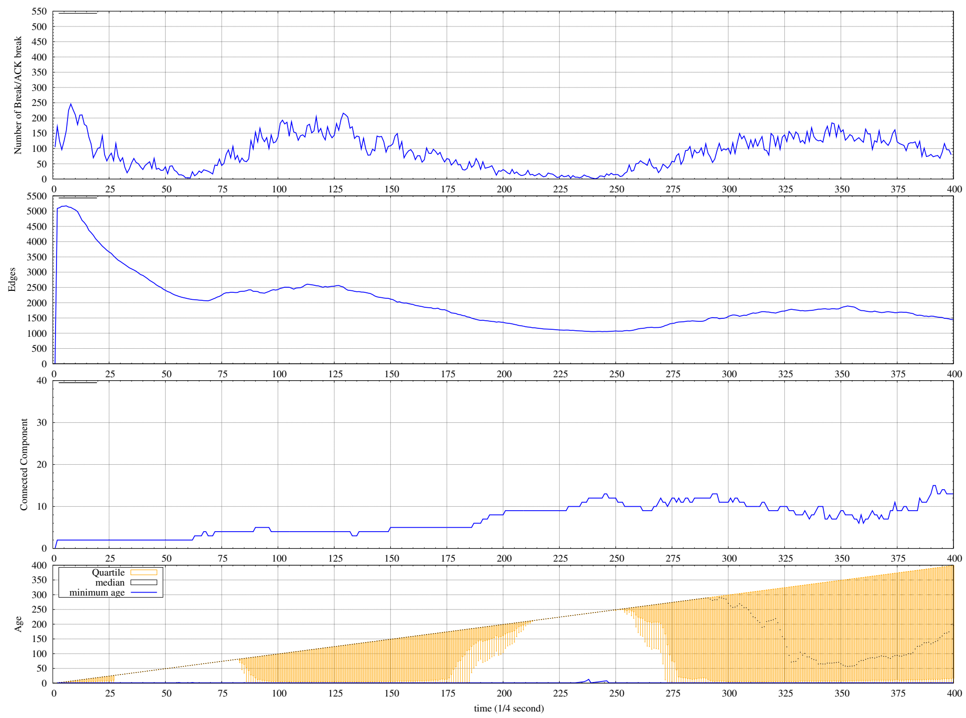


Figure 5.22: G-NODE CHANCE on Highway Configuration 4

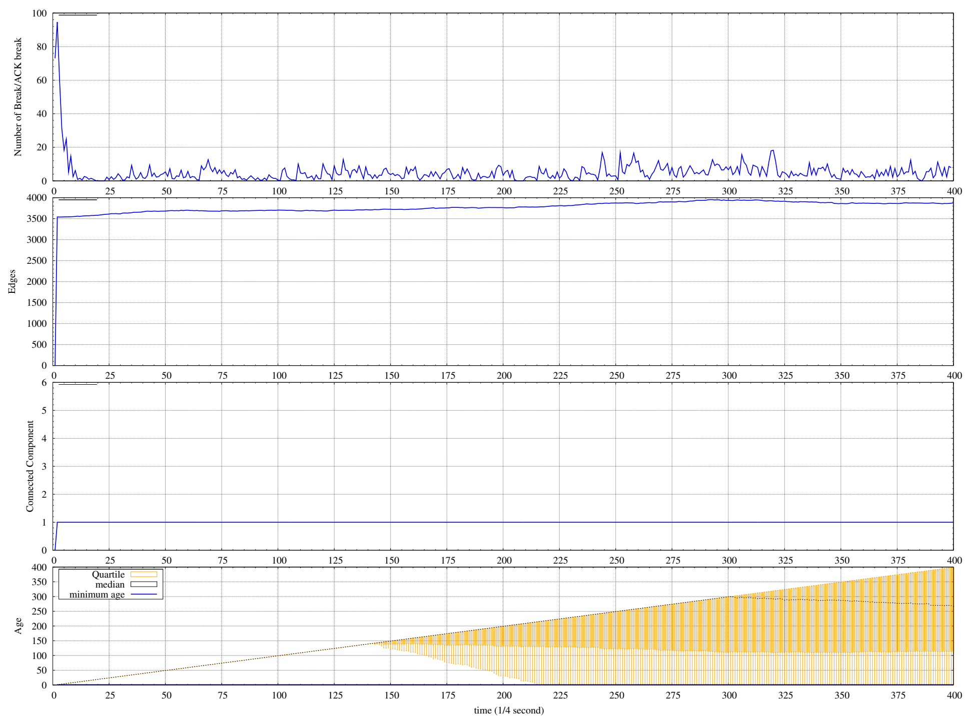
Observing closer at the beginning period of results from nodeWeight() function, the slope increasing rapidly from t_0 to t_{10}, approximately, are found especially in Shopping Mall mobility model. Within these periods, we found that CHANCE also do a lot of actions to relocate position of nodes in tree and gain a better forest solution. The more information regarding the extra number of packets used by CHANCE heuristic are gathered in Figure 5.23 for both mobility models. The Figure 5.23 compares four different characteristics which are (1) number of Break/ACK_break messages (2) Number of Edges (3) Number of connected component (4) Age of all edges at each particular time step. Both sub-figure 5.23(a) and 5.23(b) illustrate the information belong to configuration 4 of highway and shopping mall mobility models. The rest of the results can be found in Appendix A.4.

Observing at sub-figure 5.23(b), the high number of 'break' and 'ACK_break' messages during t_0 and t_{10} gives an evidence to the shift of quality found in earlier Figure 5.21. Since every nodes of the network start running the heuristics at the same time and due to the low mobility of nodes in the network, the re-locating of nodes for a better solution can be done a lot at the very beginning period. This is of course under a limitation of number of packets allowed per simulation steps which has been discussed earlier in Section 5.1.3.2. The number of connected component shows that there is no partitioning occur during the simulation time and the the changes is not substantial. More observation on age characteristic reveals that more than 75% of edges are stable (appear once and remain existing over multiple time steps). The graph of Edges show this evidence too. We can notice the higher number of edges over time. At around time step 150, the age of edges has more variation. This can be seen in box plot of age. New edges increasingly appear into the network and at the same time the number of very old age edges are reducing. However, before time step 300, the number of oldest edges are still higher than 50 percent of all edges (knowing by the median value). Therefore, the stability of edges is quite high. The increase in number of edges enables the relocating nodes but since the stability of edges after appearance is high, only small number of relocation are done. As witnessed by the number of Break/ACK_break over time, less than 20 of break and ACK_break messages have been used (except for the very beginning of the simulation).

For highway mobility model, the scale of y-axis of the Number of Break/ACK_break and Edges are fixed in order to have the similar proportion. That is the scale of Number of Break/ACK_break is ten time less than that of the Edges axis. With this similar scale, it can be seen that the overhead of CHANCE heuristic is similar to the changing of number of edges in this highway mobility model. This is true for all configurations of highway mobility models (Other results are included in Appendix A.4). Unlike the shopping mall mobility where the beginning of the simulation generates the highest number of breaking message, the overhead at the very beginning of highway mobility is not the highest peak. At this beginning period, the number of edges are so high at the beginning and the number of connected component is small. These imply that each connected component at this stage are very dense sub-graphs. One may expects the relocation to be launched a lot. However, with the limited number of packets per simulation step and the merging process of tokens by the original algorithm of G-NODE (or G-PATH), the relocation process are marginal. That is because of limited overhead at the beginning of simulation period. From the sub-figure 5.23(a), the stability of edges over time also can be observed. Approximately during time step 28 and 85, less than 25% of edges are newly appear, while 75% or more are stable edges which has the same age as the simulation step. Comparing this fact with the number of edges and number of Break and ACK_break messages, we found that the stability of communication links reduce the overhead produced by CHANCE heuristic. This also applies for the step between around 200 and 250 as well.



(a) 'highway' mobility model configuration 4



(b) 'Shopping Mall' mobility models configuration 4

Figure 5.23: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway and Shopping Mall Configuration 4

5.2.7 Efficient and Robust Topology in Multi-Objective Optimization Approach

5.2.7.1 Experimentation methodology and Implementation note

In this study, the length of simulation is 100 seconds or 400 time steps (t). Each time step graph $G(t)$ is the snapshot of the dynamic graph G at every 0.25 seconds. Figure 5.24 shows example of these snapshot graphs. Each snapshot is a graph G at moment t. At any $G(t)$, it may exist one or many connected component m. Figure 5.25 depicts a snapshot of graph at a time t emphasizing different connected components in red circles.

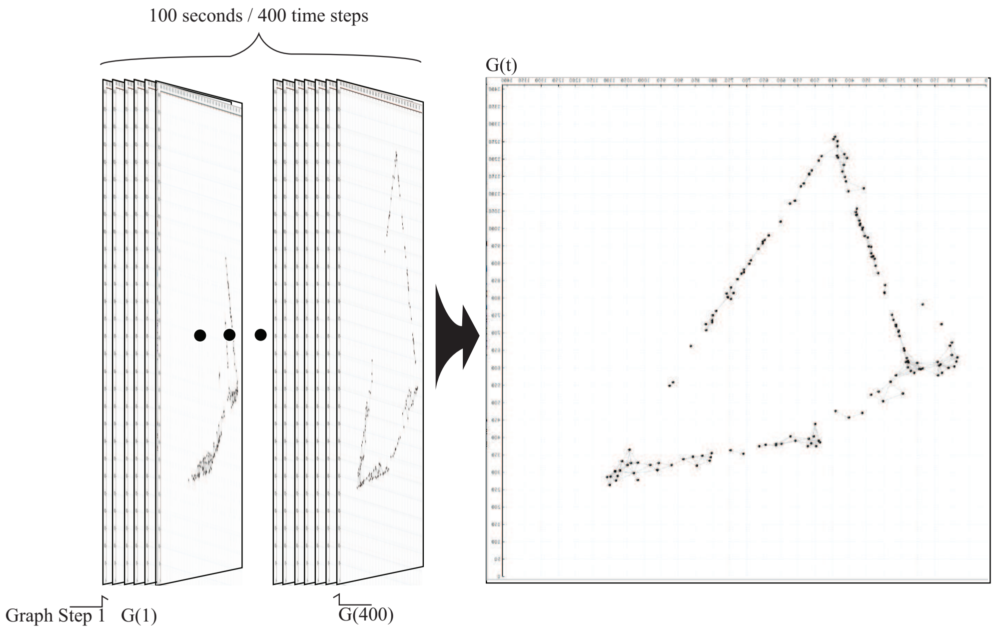


Figure 5.24: Examples of snapshots of the simulation

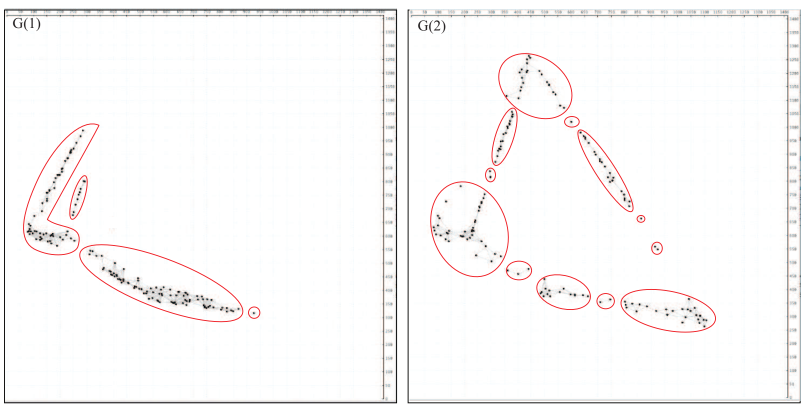


Figure 5.25: Examples of communication graph at different time step t with emphasizing of connected components in red circles

We separate each connected component of a graph G(t) and input them into jMetal one by one. The product of this process is the Pareto front of each connected component. In order to have the Pareto front of the graph G(t), all Pareto front from m connected component of G(t) are processed by addition of all cartesian products of m connected components.

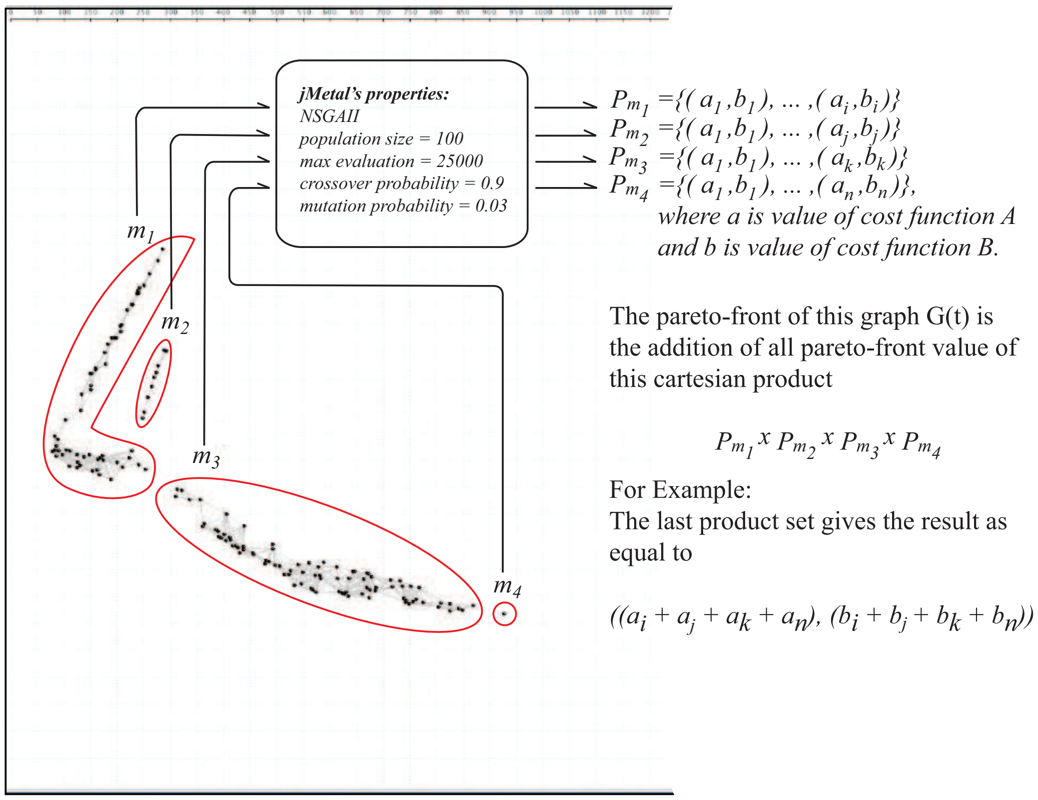
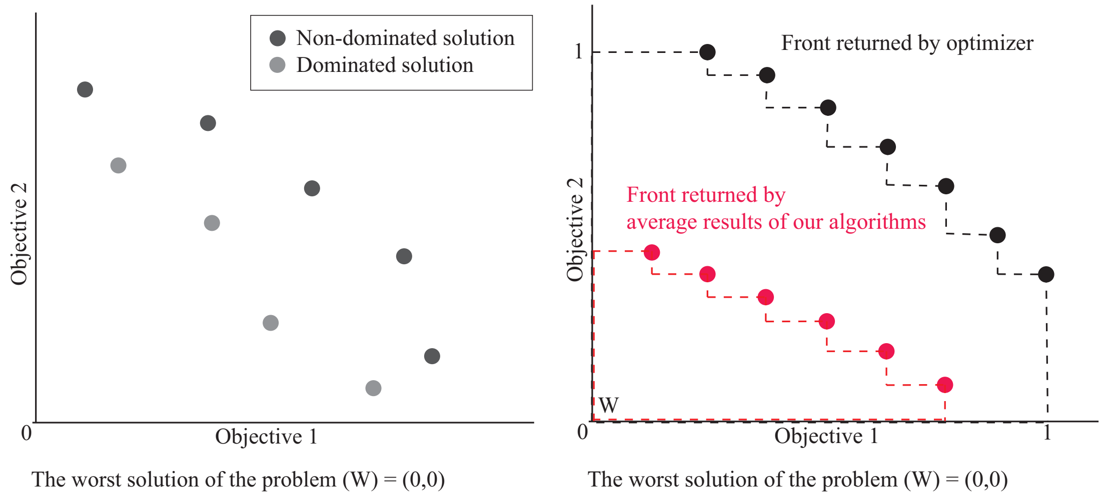


Figure 5.26: Finding Pareto front of a graph G(t)

Let M is a set of all connected component of graph $G(t)$: $M = \{m_1, m_2, ..., m_m\}$. Figure 5.25 show two different graphs, $G(1)$ and $G(2)$, where $G(1)$ has 4 connected components (M_4) and $G(2)$ has 12 connected components (M_{12}). jMetal finds the Pareto front of a connected component i, namely set P_i. In order to find the Pareto front of $G(1)$, the cartesian product of $P_{m_1} \times P_{m_2} \times P_{m_3} \times P_{m_4}$ is calculated. The value in each product set is the summation within each set. The result is used to select the Pareto front of all cartesian product again. These processes are summarized in Figure 5.26. Furthermore, parameters used for evolutionary algorithm (NSGA-II) in this work are also presented in the same figure.

In the previous sub-sections, the results are given according to a single objective problem which is concerned only one fitness value at a time. The resulting graphs are quite simple and easy to understand. However, with multi-objective problem, the solutions are more than just one value. Normally, the results are vector of solutions. In multi-objective approach, the solution space to be explored becomes n-dimensional, where n is the number of objectives to optimize. Consequently, the fitness of an individual changes from one value to set of values, each corresponding to one objective in the solution space. In such a situation, a comparison of fitness or cost function value is not sufficient as it is in single objective approach. For a multi-objective problem, a solution is rather non-dominated front or Pareto front, this definition is given earlier in Section 2.2.2. Therefore, the assessment of quality must be done for a whole group of solutions at the same time. Hypervolume (HV) is used here to assess the quality of a whole group of solutions at the same time. The motivation for using HV is for comparing between solutions given by jMetal (Pareto front given by meta-heuristic, NSGA-II, in this case), DA-GRS reference algorithm and our proposed heuristic. Hypervolume has been discussed in details in [139]. We present here briefly the methodology to obtain HV using maximization problem as an example.



(a) Pareto front and dominated solutions (b) hypervolume of a Pareto front

Figure 5.27: Multi-objective search space

Figure 5.27(a) shows the relation between a set of solutions in gray color that is dominated by another set of solutions, in dark gray color. In another figure, the Pareto front given by optimizer is in black color. All solutions in this particular figure has been normalized by the value of Pareto front so that the maximum value of each objective reach one. The hypervolume of Pareto front can be obtained by calculating the area under the black dotted line. Please note that this example is for maximization problem which has the worst solution point at $(0, 0)$. In the next section, the comparison has been made between the HV of Pareto front given by NSGA-II and the average result of our proposed algorithms at different preference of trade off value, ρ. In Sub-Figure 5.27(b), the set of front in red color represent a set of results to compare with Pareto front. The robustness metrics are also presented to measure how well G-Node-Path handle the construction of spanning forest at different preferences, ρ.

For ease of reference and comparison of result, the preference of trade off value, denoted as $[\rho, 1-\rho]$, is set for the experimentation as shown in Figure 5.28. So, there are eleven different experimentation for each configuration network. Each experimentation in each configuration network uses the preference trade off value for heuristics [G-NODE, G-PATH] accordingly.

This work implements the multi-objective spanning tree and all evolutionary operation corresponding to Edge-Set representation [135] into jMetal.

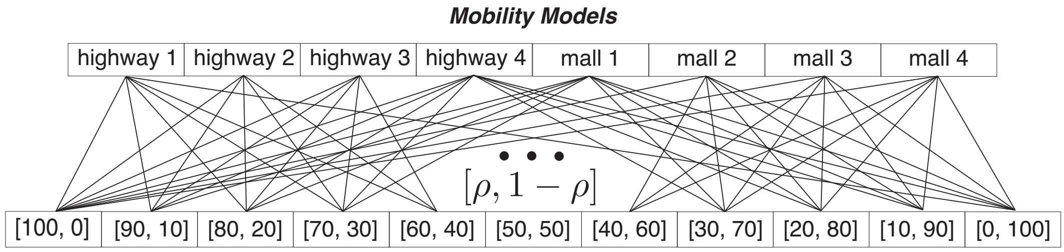
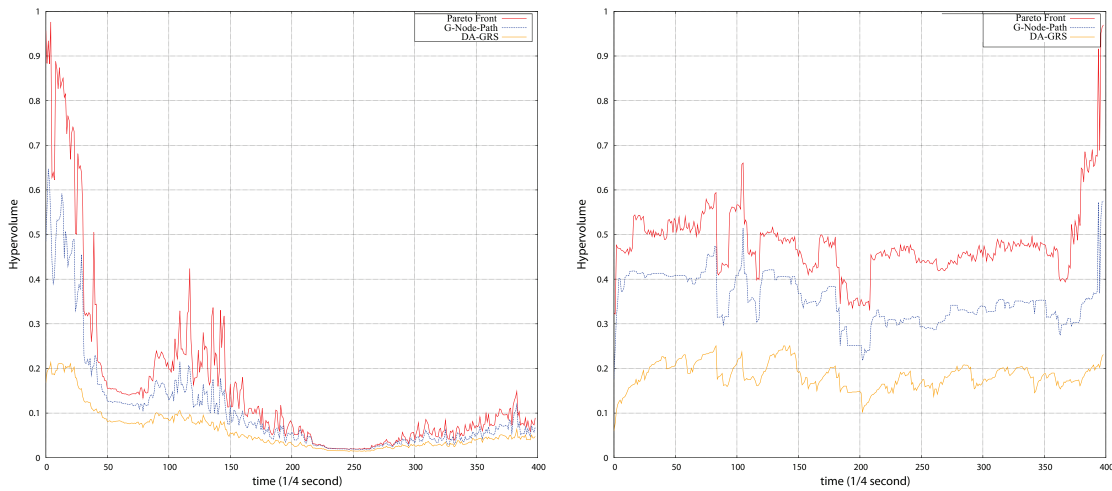


Figure 5.28: The preference trade off value, ρ, and the experiment methodology

5.2.7.2 Hypervolume results

The comparison between the hypervolume of Pareto front and other heuristic results give us the quality of local decision making heuristic against the best solution when global information is provided. In this section, different graphs show the comparison of hypervolume between Pareto front given by NSGA-II, DA-GRS and our G-Node-Path heuristic. The experimentation has been done using 100 runs in each configuration of network. Please note that all resulting in this section can be found in Appendix A.5.



(a) Highway mobility model

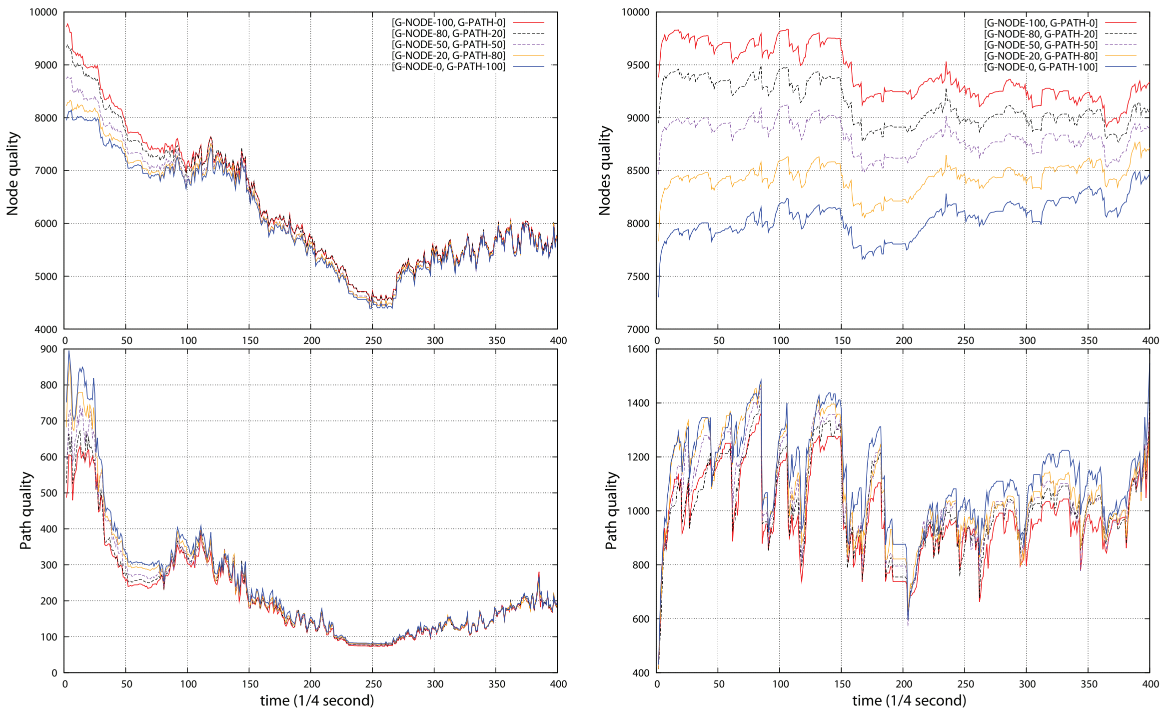
(b) Shopping Mall mobility model

Figure 5.29: Comparison between hypervolume of NSGA-II, DA-GRS and G-Node-Path over simulation time

The result graphs 5.30 reveals and advantage of G-Node-Path. It is clearly see that G-Node-Path algorithm achieves more than a half of distance between Pareto front and DA-GRS on both mobility models. Both sub-figures also depict the strong differences of mobility characteristics over simulation time. Focusing at Sub-Figure 5.29(a), between time step 300-350 where the topology is highly fluctuant, G-Node-Path still provide a better solution. In summary, these results confirm the benefit of G-Node-Path for handling multi-objective topology of spanning forest in DT-MANETs. Next sub-section we shall discuss the results based on different metrics that form problem description at different preference value, ρ.

5.2.7.3 Robustness Metrics Results

The comparison between different preference value is used to confirm the usability of G-Node-Path with preference value. At Figure 5.30, the overall output topology is influenced by the preference value, and thus, each metric value presents this fact accordingly. Figure 5.30 has two sub-figures for presenting results from Highway and Shopping mall mobility model. In each sub-figure, there are two different graphs which illustrate node and path quality value over simulation time and are given as a result of different preference value (ρ) of G-Node-Path heuristic. Each information line can be distinguish by color and corresponding key such as [GN-100, GP-0] which means this vector of information are generated by G-Node-Path at 100% of G-NODE and 0% of G-PATH. The path quality used here is the capacity bandwidth and the nodes' quality is the battery level. The experimentation has been done using 25 runs for each configuration of network. Please note that all resulting graphs from this section can be seen in Appendix A.6.



(a) Highway mobility model

(b) Shopping Mall mobility model

Figure 5.30: Comparison between path quality and node quality of five different preference value (ρ) over simulation time

In Sub-Figure 5.30(b), it can be seen clearly that each set of experimentation are distinguishable quite well, especially for nodes' quality. Although, the differences between each preference set is quite slim for path quality and also for highway mobility model, the pattern of differences can be seen.

5.3 Summary

In this Chapter, the experimentation methodology including assumptions, characteristics of the underlying network used in this study are elaborated and verified. This study focuses on two different mobility models which are highway and shopping mall mobility models derived from Madhoc simulator. All the simulations are done using GraphStream simulator. We introduced two communication boundaries for GraphStream based on Gupta's finding, found at Equation 5.1, and the beaconing rate of IEEE802.11, Local Area Network standard [103]. These communication boundaries are useful in dynamic simulation of communication graph. They force the end of time step and trigger the next step to be launched in the simulation.

Results and analysis of each algorithms and heuristics introduced in this study are presented and extensively analyzed. In summary, we found that results confirm the benefit of greedy heuristic used by G-NODE and G-PATH, both of them are fully decentralized algorithm and use only local information. By using only one-hop information, G-NODE and G-PATH achieve a better results than DA-GRS algorithm but not yet reach the optimality. The reason is G-NODE and G-PATH do not possess the change management methodology. BREAK heuristic is implemented for re-organizing the existing solution based on a different setting time frame (T_break). The mechanism is simply to break away those nodes who have quality lower than a threshold periodically and leave the connecting process to G-NODE once again. Although, the results of G-NODE BREAK show the improvement on the robustness of constructed spanning forest periodically (when it is the time of T_break), the over time robustness is decreasing. CHANCE heuristic takes the benefit of overhearing '$findingTk$' message from a node possessing a token. Immediately, nodes (who are in any tree and do not possess token) compare the quality $QS_{current}()$ with its parent (the parent node notifies the direction of token in a particular tree) and the quality $QS_{chance}()$ with the node broadcasting $findingTk$ message. If $QS_{chance}()$ is higher, the node breaks with its parent node and enter as a choice for merging with the '$findingTk$' broadcaster. Results show that G-NODE CHANCE almost reach the optimality in single-objective problem.

The multi-objective problem presented in this Chapter is the combination of two objectives problem concerning the quality of node and edge. This study implemented the G-Node-Path as a distributed and localized algorithm for constructing spanning forest topology based on bi-criteria. Results show that the set of solutions provided by G-Node-Path can achieve different level of desired solutions. Furthermore, the hypervolume of the solutions set provided by G-Node-Path achieve a better solution than DA-GRS reference algorithm.

Chapter 6

Conclusions and Perspectives

Contents

6.1 Conclusion **141**

6.1.1 The Framework 141

6.1.2 Robust Quality Criteria 141

6.1.3 Solution Algorithms and Heuristics 142

6.1.3.1 Token traversal strategy 142

6.1.3.2 Construction and maintenance process 142

6.1.3.3 Re-organization process 143

6.1.4 Evaluation Methodology 143

6.2 Perspective **144**

6.1 Conclusion

Delay-tolerant mobile ad hoc networks, shortly denoted as DT-MANETs, has become an emerging relative class of mobile ad hoc networks (MANETs) that feature frequent and long-duration partitioned MANETs. DT-MANETs is a challenged environment of MANETs where end-to-end connectivity cannot always be obtained. Such environment can be compared as a disconnected or partitioned MANETs. Any protocol that can withstand this partitioned feature works also in a connected network of MANETs. This study models DT-MANETs using graph theory. Dynamic communication graph is used to present mobile nodes and available communication link between nodes at a given moment t.

Generally, topology management (TM) is a technique to control the network topology in order to provide better control over network resources and to increase the efficiency of communication. To overcome some difficulties, topology management selects a deliberate choice of a subset of active links or active nodes or both to restrict a topology of the network. This study focuses on Topology Management problem in DT-MANETs. The spanning forest topology is selected for TM in this study due to its strong benefit for operation in networking and its popularity used as internal process of other network protocol. In particular, we aim at selecting spanning tree such that avoid the deterioration of network, or increase the quality of service in DT-MANETs by utilizing different quality factors existing in each communication node and edge.

6.1.1 The Framework

This study aims at providing a framework of TM that comprises of (1) a set of robust criteria for topology management problem, (2) a set of solution algorithms that are decentralized, asynchronous algorithms which utilize local knowledge, and (3) evaluation methodologies for different solutions of different problems (e.g., single-objective and multi-objective problem). The framework are divided into three main sectors according to its components stated above. The robust criteria selected by this study are the trust level of cooperative enforcement paradigm, the available energy of entity and the capacity bandwidth of communication links. The combination of such criteria together for robust topology management problem is a novelty which has not been dealt with in literature to date. In the solution phase, this study implements G-NODE, G-PATH, G-Node-Path, DFSmove, BREAK, and CHANCE heuristic as solutions of the framework. These algorithms and heuristics are totally decentralized and use only one-hop information. The decision is making locally without any consultation with other nodes. In the final phase of this framework, simulation programs and methodologies are prepared for evaluating those proposal algorithms based on different aspects to the problem. The evaluation phase is done offline and global information is assumed. The framework can be summarized into graphical chart as shown in Figure 4.23.

6.1.2 Robust Quality Criteria

Qualities of node and edge also influence the quality and robustness of topology structure. Within the framework, three different quality criteria have been chosen due to their ability to enhance the quality of the overall structure at different constraints. Since ad hoc networks rely on cooperation of a set of nodes in order to emerge and operate the network, our first quality criterion is trust and reputation level from cooperative enforcement aspect. The high trust level of a node means the node collaborates efficiently to maintain the network. On the other hand, selfish node can deteriorate the robustness of the network because they deny forwarding a packet received from others. The second quality criterion is the remaining battery level in node. The ability of mobile node to operate further in time is one of the most important characteristics in ad hoc network. The third quality criterion aspect is the capacity bandwidth available in each communication link. The bandwidth available for communication of the nodes sharing a common broadcast radio channel is usually limited and restrained. The capacity bandwidth of each communication edge gives us an immediate idea of how well the edge prompts to transfer data.

Hence, robust spanning forest should comprises of high quality nodes positioning as forwarder (having multiple tree degree) and the low quality nodes should be located at leaves since loosing it at this point has less effect to the efficiency of the overall topology. Furthermore, less capacity links should not be chosen as virtual backbone link (tree link) since their existence of low capacity in tree topology can create a bottle neck at the end-to-end communication level.

6.1.3 Solution Algorithms and Heuristics

6.1.3.1 Token traversal strategy

Token management approach for constructing and maintaining spanning forest often comprises two processes, which are the traversal of token within a tree and the trees merging operation (construction and maintenance process). The framework proposed new algorithms to both process, as well as the re-organization process module.

The utilization of token is a common mechanism in spanning tree algorithms. One process of token management is the traversal process of the token itself. This study introduces a DFSmove, a depth-first-search heuristic, as a token traversal strategy in token management paradigm. Our proposed DFSmove is purely decentralized and uses memory storage on nodes to ensure its deterministic nature. Actually, this deterministic nature of DFSmove can influence a faster spanning of tree in distributed environment. To the best of our knowledge, this is the first time depth-first-search heuristic is introduced and used for such purpose and it outperforms previously known technique (this has been affirmed by simulation).

6.1.3.2 Construction and maintenance process

Greedy Robust Spanning Tree or G-NODE is a decentralized algorithm that employs greedy algorithm for selecting a high quality node to operate with (trees merging process). Also this study implements G-PATH algorithm which has the similar characteristics as G-NODE (utilizing greedy heuristic), but it is concerned with the quality of communication edges. Both algorithms improve the careful rule-based token management of DA-GRS reference algorithm by relaxed synchronization method, enhancing token traversal strategies (employ our DFSmove), enhancing merging operation and only utilized one-hop information. The information gathering is done by periodically sending message (i.e. the existing beacon). Indeed, the communication protocol and details are introduced to DA-GRS reference algorithm in this study.

G-NODE and G-PATH focus on one particular quality of entity at a time and thus they are used to solve single-objective problem (i.e., when the desired topology is based on one particular characteristic). However, the nature of topology management problem is a multi-objective one. A multi-objective problem is problem with multiple criteria that are in conflict with each others. G-Node-Path algorithm is introduced into this study to provide solutions to such multi-objectives problem for spanning forest topology. By combining G-NODE and G-PATH together, G-Node-Path uses preference value, given offline, of each criterion to influent the selection between G-NODE or G-PATH at one moment of each node. It is worth to emphasize that it works in purely decentralized and using only local knowledge (one-hop information).

6.1.3.3 Re-organization process

Although, G-NODE and G-PATH surpass DA-GRS reference algorithm, the results can still be improved since there still are a big distance between optimality and the solutions provided by such decentralized algorithms in terms of our robustness metrics. Moreover, we found the fact that the robustness results tend to decrease over time. Our analysis leads to the conclusion that G-NODE and G-PATH can benefit from an adaptive management process for the existing tree/forest. The greedy heuristic implemented in this study are very useful when many new nodes turn on together into the system, and when devastating situation happens, so the topology is severely scramble. In other words, greedy heuristic of G-NODE and G-PATH are effective when they are multiple tokens meeting together. However, changes that occur after the forest established cannot be effectively managed.

In order to maintain effective management, two different improvement on adaptive re-organization process are implemented. BREAK and CHANCE heuristics have different approaches. BREAK heuristic uses periodical approach for checking current situation (topology and surrounding) of each node and relocate the tree connection in decentralized way, if need. BREAK uses a number of overhead messages for relocating task. However, its mechanism is too simplistic so the results only achieve a short term improvement over time. This is also depending on how often the checking period is performed. However, BREAK heuristic is very useful at the beginning period of topology construction in low mobility network. BREAK can correct the fault decision happened by greedy heuristic and obtain a better solution for a longer period due to low changes of the network. Unlike the BREAK heuristic, CHANCE is implemented as an opportunistic based heuristic. It does the relocation of topology only when it is advancetegeous. The CHANCE heuristic is proven to be very interesting since it reduces the gap from optimality occurred by using G-NODE or G-PATH alone. Indeed, CHANCE also utilize some overhead messages for relocating tasks. However, it is proven by our simulation that such overhead are not effected the overall communication of the network.

6.1.4 Evaluation Methodology

The evaluation methodologies of our framework are divided into two modules. One is for single-objectives problem which provides few common exact algorithms for finding optimum solution and also other statistical measurements. For multi-objectives problem, we equip our framework with jMetal, Meta-heuristic algorithms in java-based framework. A proper set of operations for multi-objectives evolutionary approach is implemented in this study into jMetal for spanning forest problem.

In summary, topology management problem in DT-MANETs must be tackle in decentralized, asynchronous and utilizing local information manner as much as possible. These requirements are well-known in DT-MANET context. This study focuses on constructing and maintaining efficient and robust spanning forest topology in particular as a mean to topology management. A number of algorithms and heuristics which comply to the constraint stated above are proposed. Some combinations of these proposed algorithms and heuristics are proven by simulation by resulting in robust spanning forest topology in single-objective problem (close to optimum). G-Node-Path algorithm can provide a wide range of solutions for multi-objectives problem. Using a preference value set, G-Node-Path can provide a solution according to desired properties.

6.2 Perspective

Future works can be carried out in many aspects as follows. Exploring the opportunity of automatic tuning of G-NODE BREAK (i.e. G-NODE with BREAK heuristic) is interesting. This may result in providing nodes with ability to adapt and learn from their experience and local knowledge. This will enable nodes to determine their own suitable next T_break even under the assumption that global knowledge cannot be assumed. The idea of machine learning should also be provided for fine-tuning the robustness of multi-objectives topology over time.

This study provides robust spanning forest topology by selecting a topology that increase the quality of service in DT-MANETs by utilizing different quality factors existing in each node and edge. Another approach to provide a robust topology is to provide an alternative topology such that even one path disrupted, the alternative path can operate instead. Maintaining multiple coloring spanning forest at the same time as providing the backup route will be advantageous. The multiple coloring forest idea is not novelty but the construction, maintenance and swapping among possible trees in a fully decentralized way and with the least control messages possible will enhance this approach to topology management considerally.

According to the validation of this study, the simulator is an abstract one. It will also be very beneficial to implement the proposal solutions and validate them using testbed scheme and/or using other complex simulators (e.g. hardware layer simulator such as NS-3, OpNET). The result from testbed and complex simulators are valuable and may reveal further idea and directions for improvement.

Although this study interested in different mobility model, the stability issue was out of scope to this study. However, it will be very interesting to focus on the characteristic of both mobility and the stability of connection among nodes too. Understanding the stability characteristic may be used to reduce the unnecessary topology management operation. Thus, some operations regarding stability statistical measurement may be useful to implement in solution algorithm. Following the above idea, the strong connectivity entities (community/-group of nodes) is also an interesting issue. The connectivity criterion may be used to define the community/group boundary and scope the size of topology management down to necessary areas by disregard some non-community path. Within such a strong community, our topology management using tree-based topology approach will contribute the robustness on connectivity among nodes within a community. Furthermore, the communication between inter-community should also be studied extensively to enhance the overall robustness among multiple groups/communities.

In general, time and message complexity used by the decentralized algorithms in dynamic environment like DT-MANETs are not really relevant to the current study and also out of the scope of this study. However, from theoretical point of view, further study may be beneficial. A theoretical or mathematical proof of the spanning forest convergence under dynamic scenario by the proposal algorithms presents a real challenge.

Since spanning tree is used as a pre-processing method of many research works for obtaining other topology (i.e., connected dominating set, cluster-based topology), it is interesting to apply the decentralized algorithm provided by this study in other application contexts.

Part II

Appendix

Appendix A

Results in Graph

Contents

A.1 Robust topology utilizing Cooperative Enforcement Approach . 148
A.2 Robust topology utilizing Energy Availability Approach 152
A.3 Robustness and Efficiency Results for CHANCE heuristic 156
A.4 Overhead Messages for CHANCE heuristic 160
A.5 Hypervolume Result 168
A.6 Robustness Metric Result for Multi-objective problem 170

A.1 Robust topology utilizing Cooperative Enforcement Approach

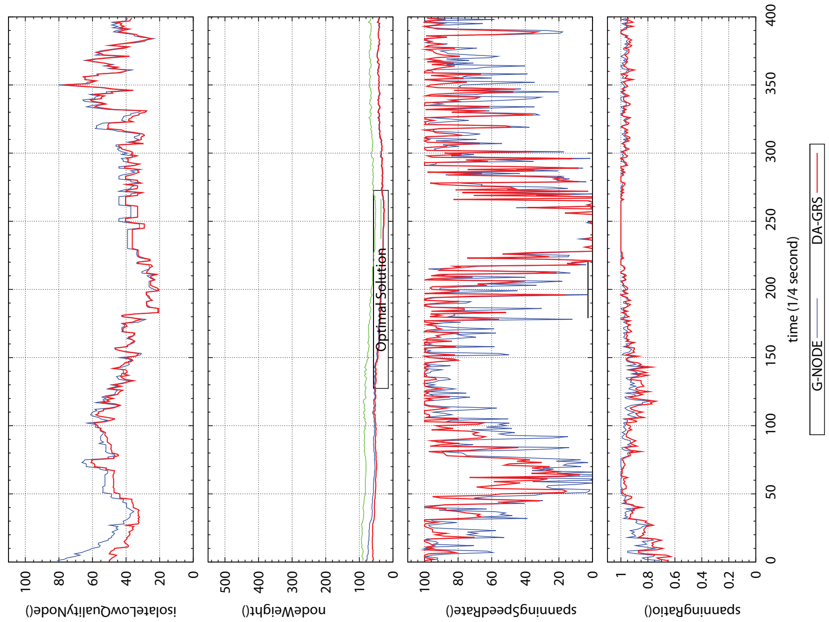


Figure A.1: Cooperative Enforcement Approach: G-NODE on Highway Configuration 1

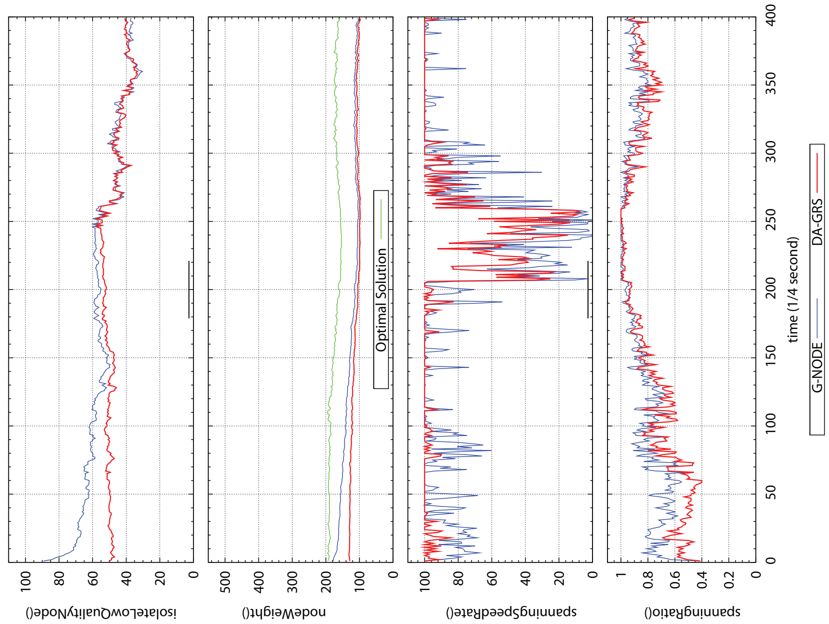


Figure A.2: Cooperative Enforcement Approach: G-NODE on Highway Configuration 2

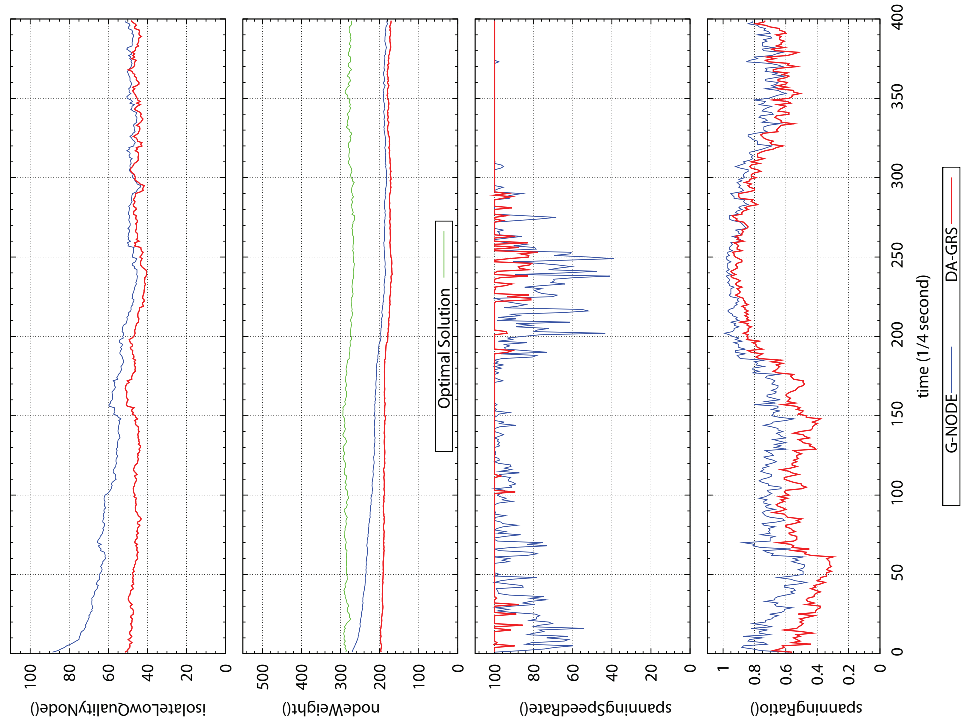


Figure A.3: Cooperative Enforcement Approach: G-NODE on Highway Configuration 3

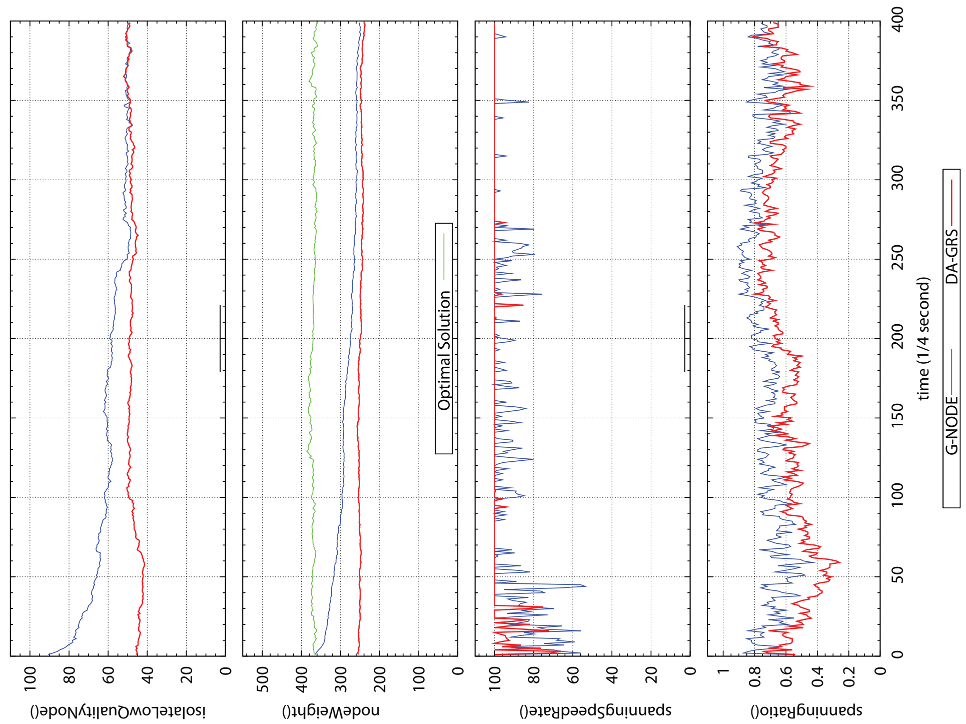


Figure A.4: Cooperative Enforcement Approach: G-NODE on Highway Configuration 4

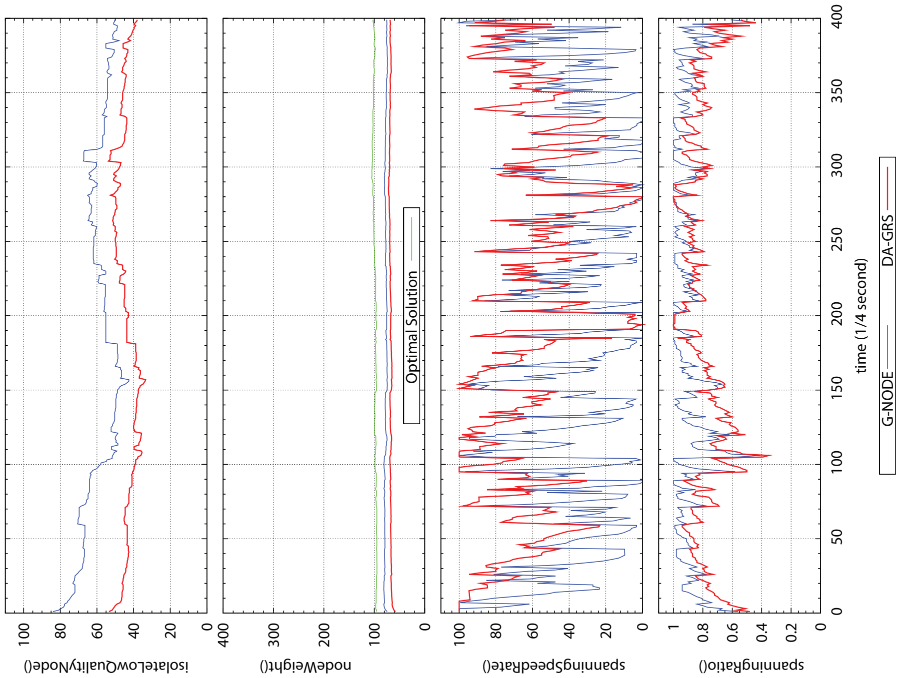


Figure A.5: Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 1

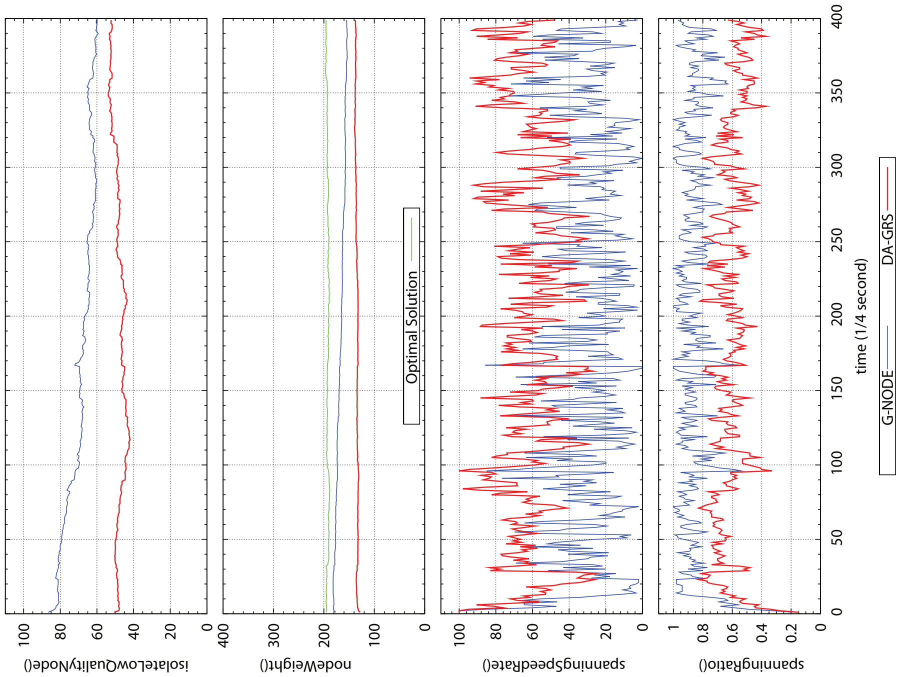


Figure A.6: Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 2

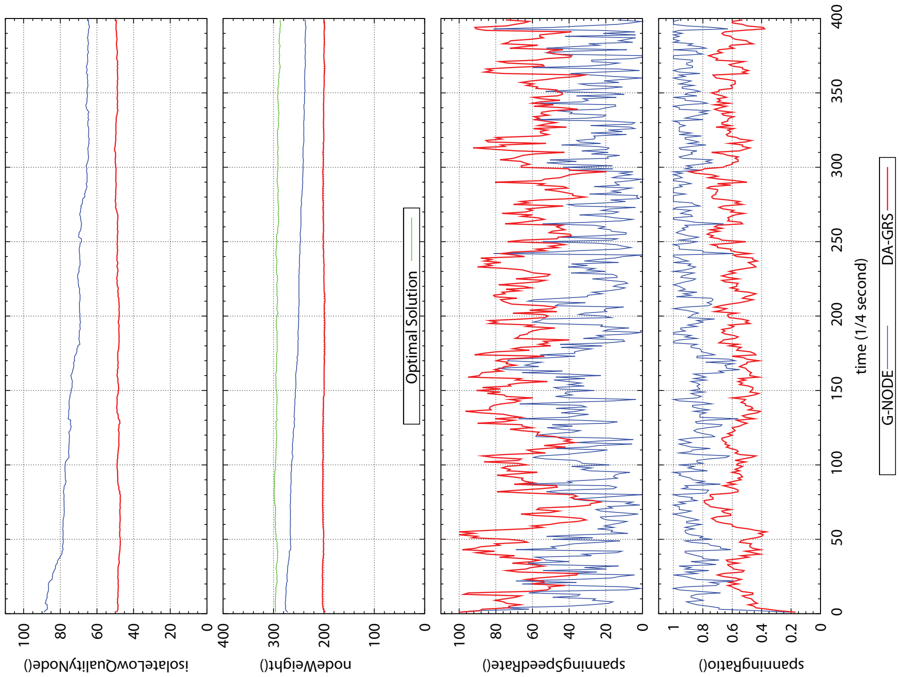


Figure A.7: Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 3

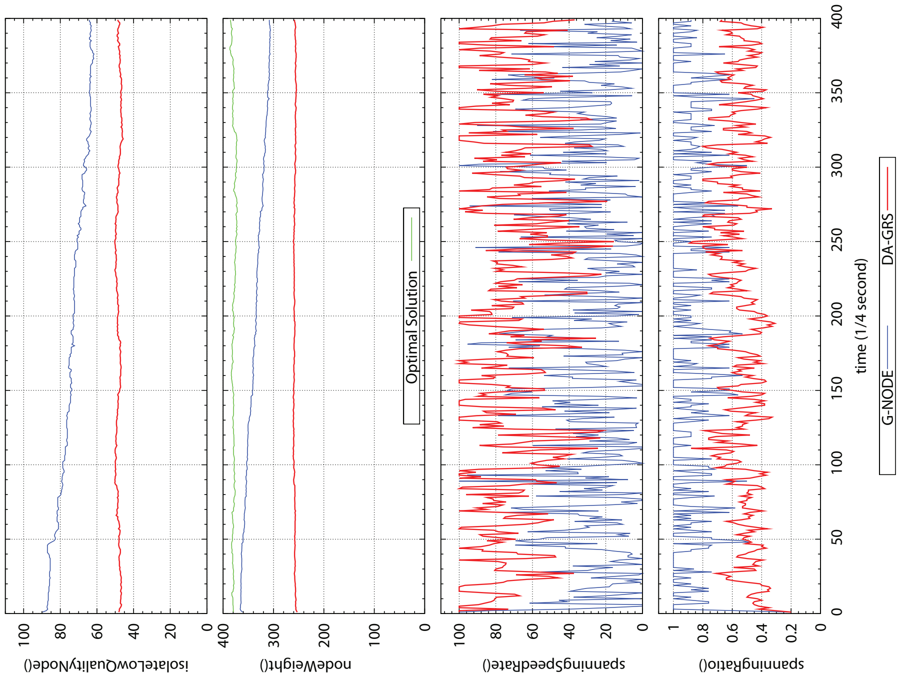


Figure A.8: Cooperative Enforcement Approach: G-NODE on Shopping Mall Configuration 4

A.2 Robust topology utilizing Energy Availability Approach

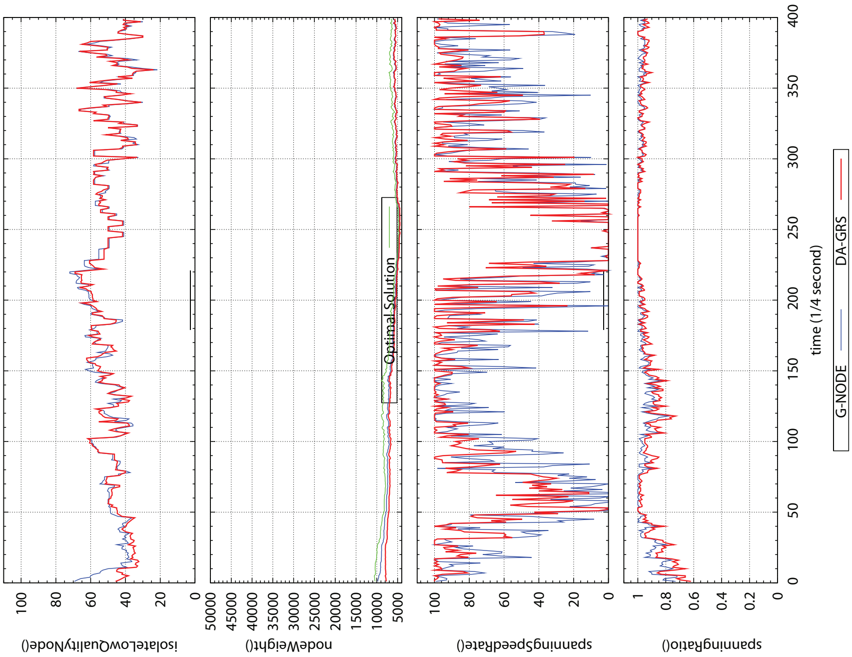


Figure A.9: Energy Availability Approach: G-NODE on Highway Configuration 1

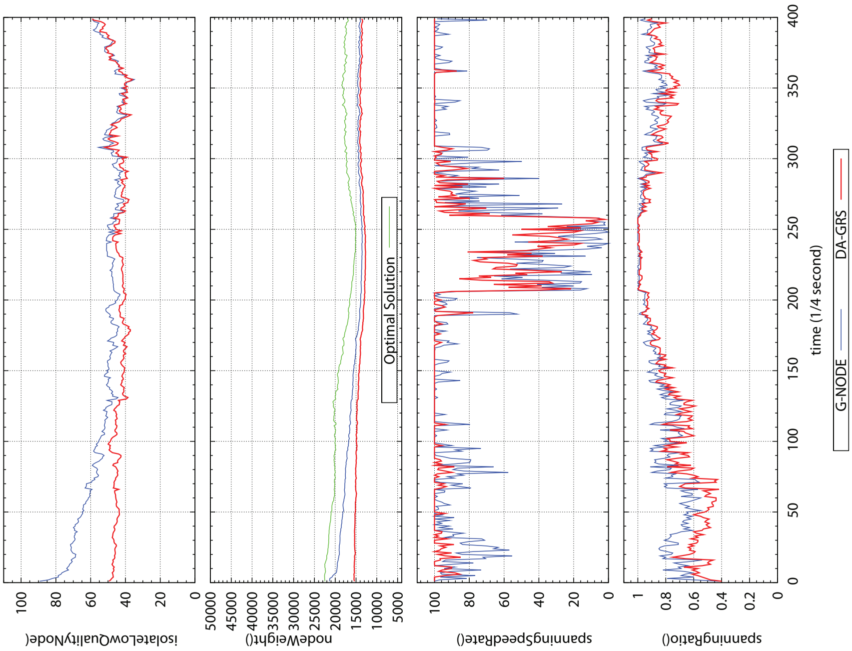


Figure A.10: Energy Availability Approach: G-NODE on Highway Configuration 2

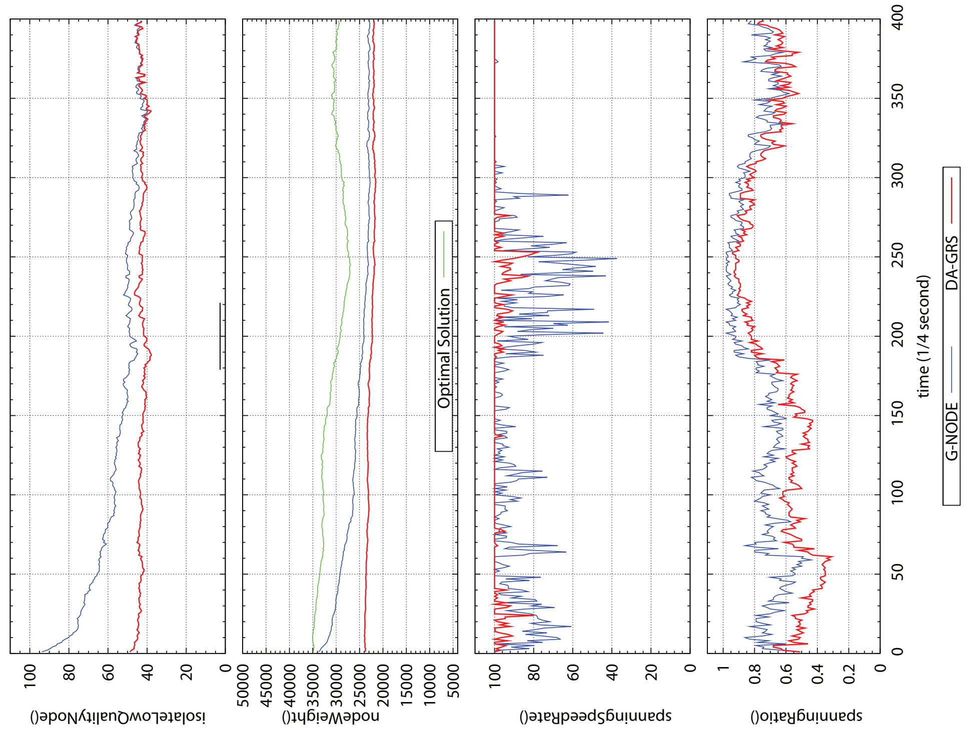


Figure A.11: Energy Availability Approach: G-NODE on Highway Configuration 3

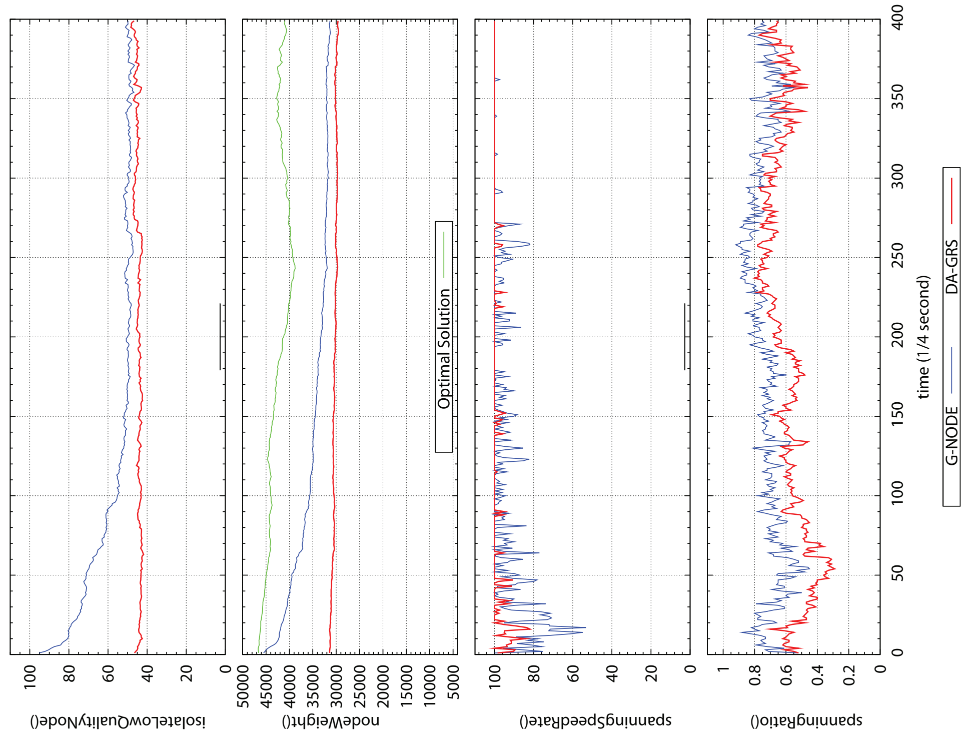


Figure A.12: Energy Availability Approach: G-NODE on Highway Configuration 4

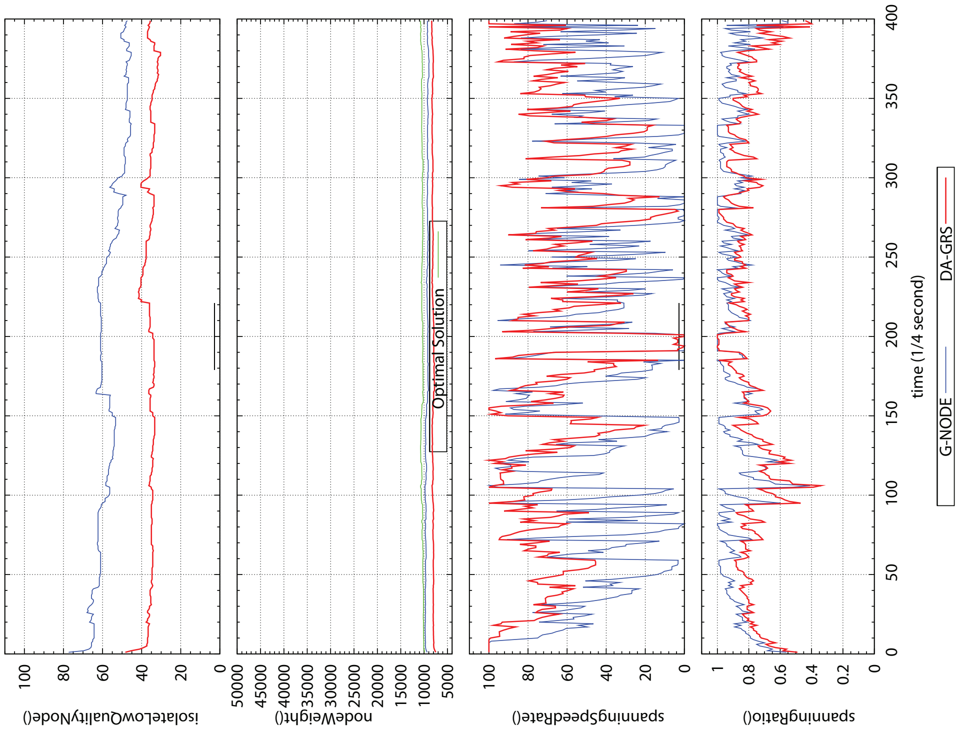


Figure A.13: Energy Availability Approach: G-NODE on Shopping Mall Configuration 1

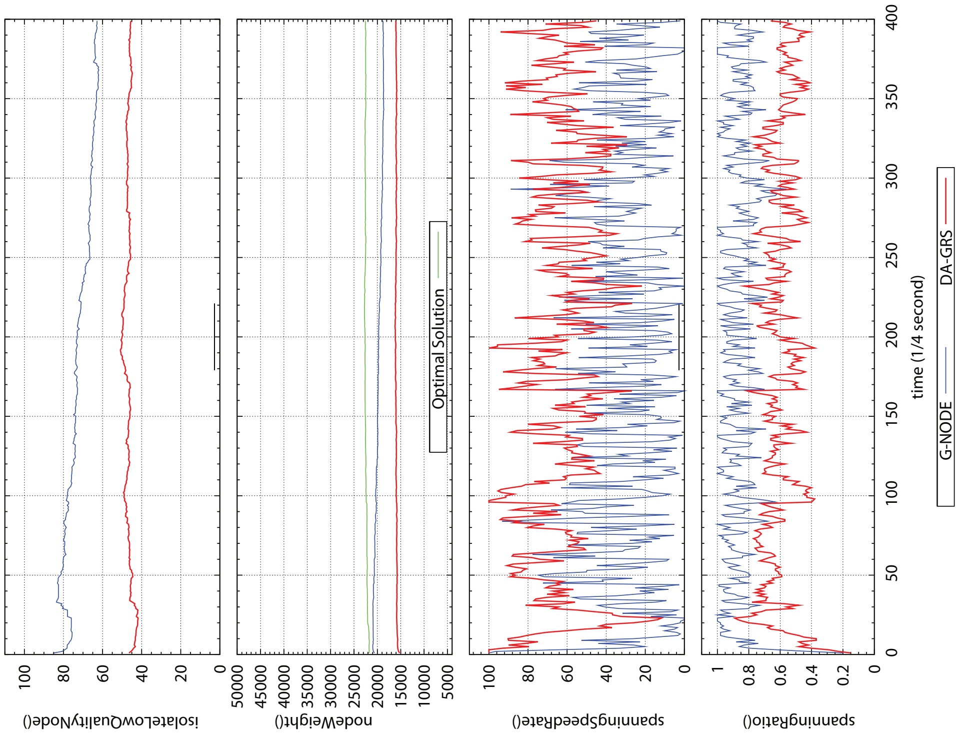


Figure A.14: Energy Availability Approach: G-NODE on Shopping Mall Configuration 2

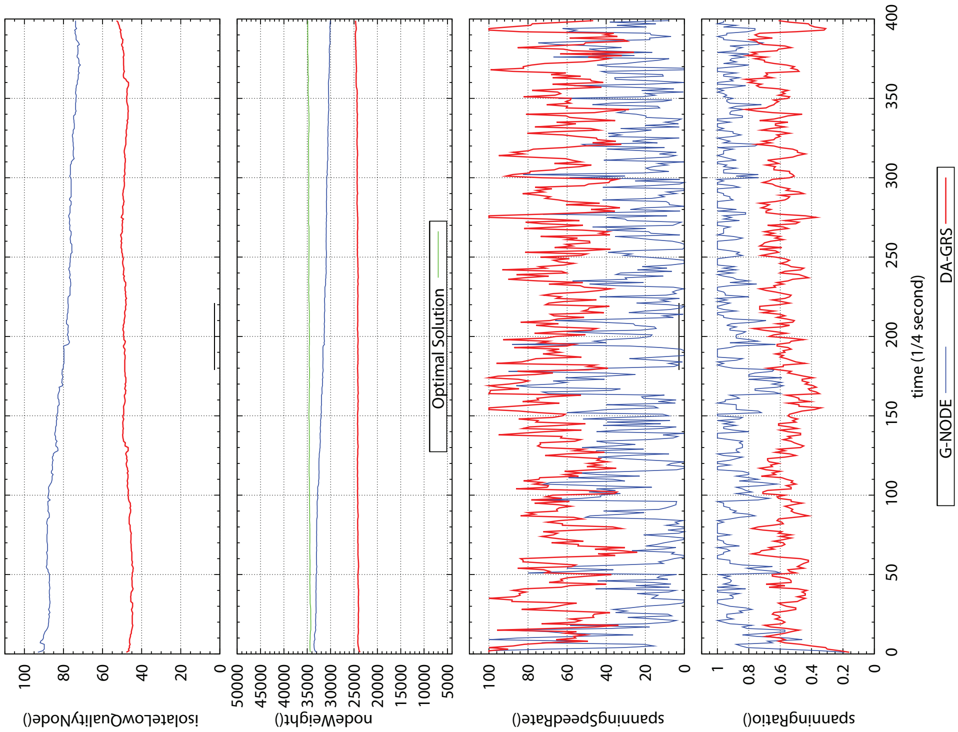


Figure A.15: Energy Availability Approach: G-NODE on Shopping Mall Configuration 3

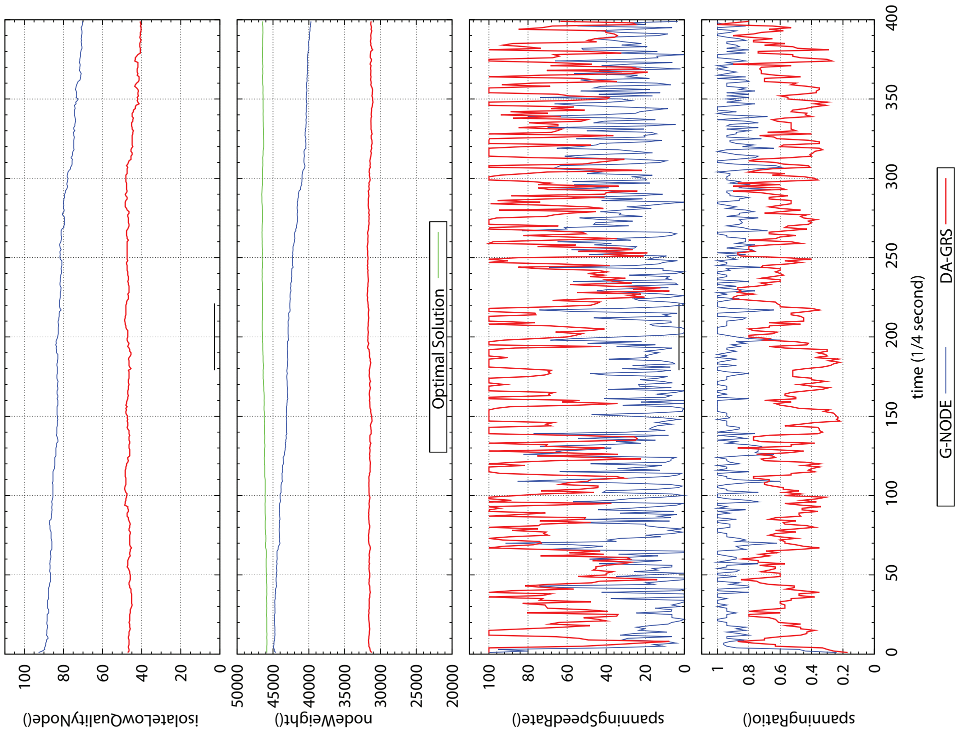


Figure A.16: Energy Availability Approach: G-NODE on Shopping Mall Configuration 4

A.3 Robustness and Efficiency Results for CHANCE heuristic

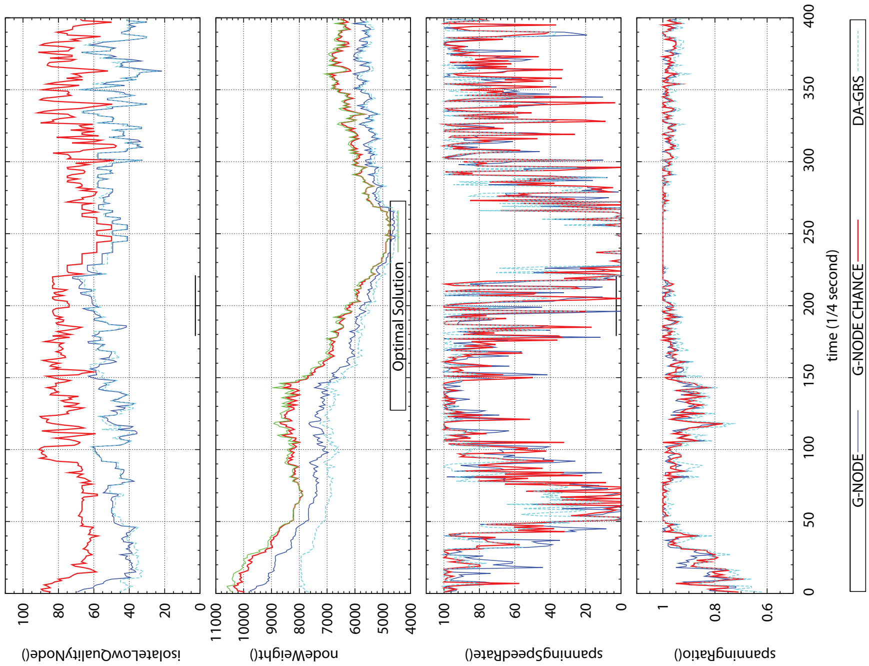


Figure A.17: G-NODE CHANCE on Highway Configuration 1

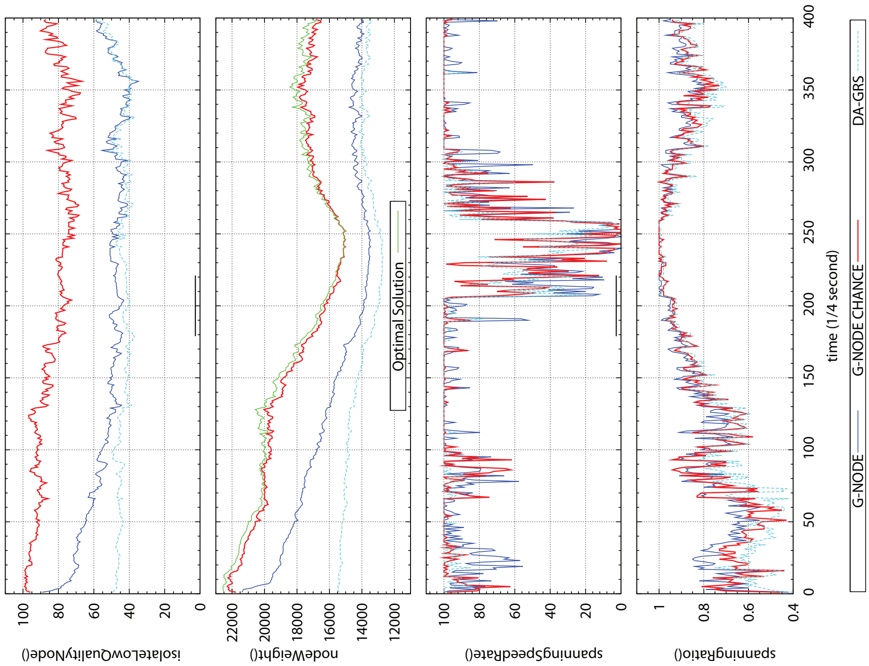


Figure A.18: G-NODE CHANCE on Highway Configuration 2

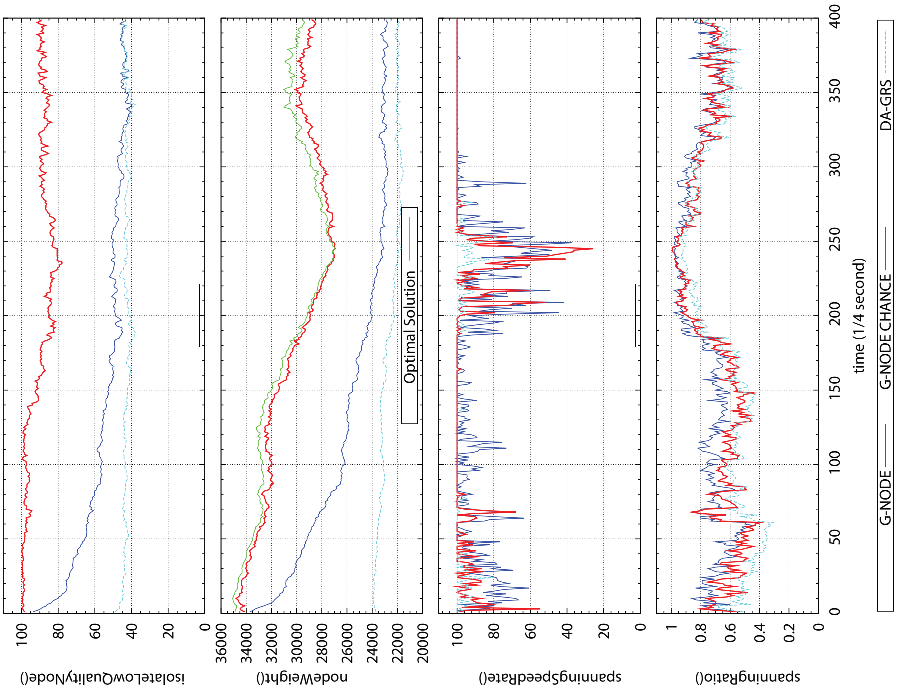


Figure A.19: G-NODE CHANCE on Highway Configuration 3

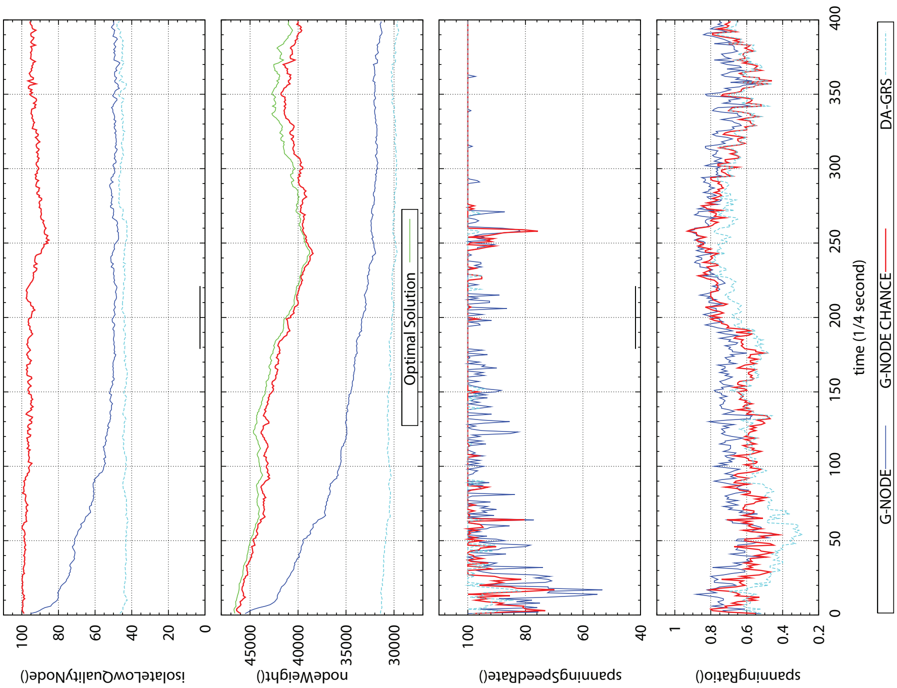


Figure A.20: G-NODE CHANCE on Highway Configuration 4

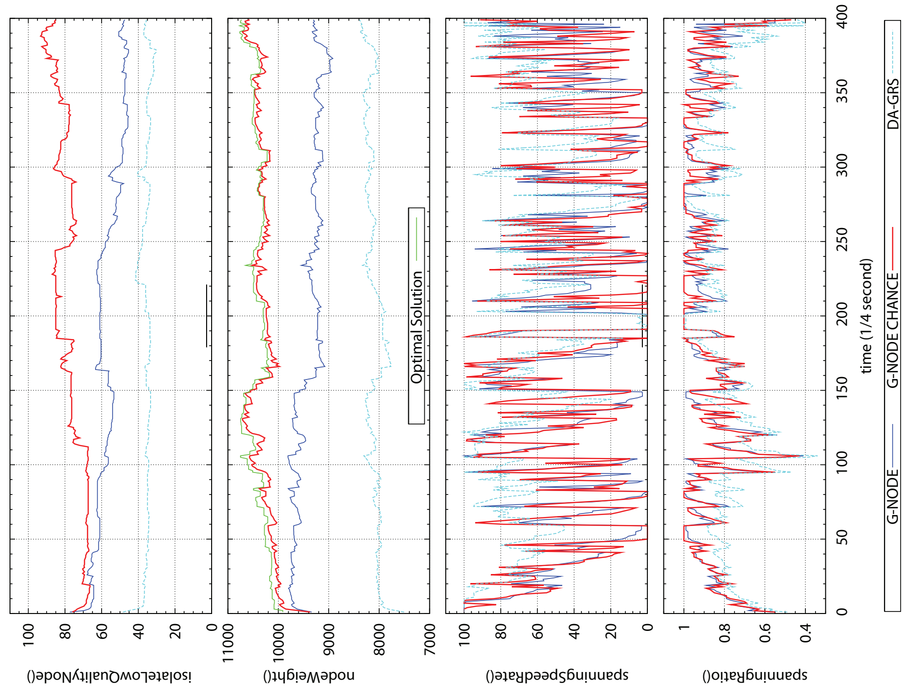


Figure A.21: G-NODE CHANCE on Shopping Mall Configuration 1

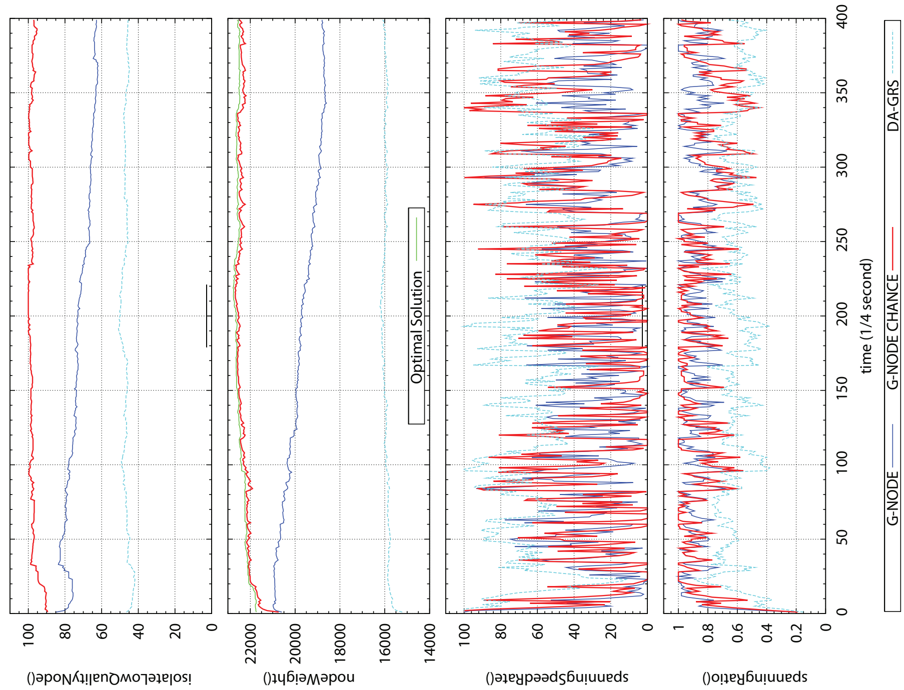


Figure A.22: G-NODE CHANCE on Shopping Mall Configuration 2

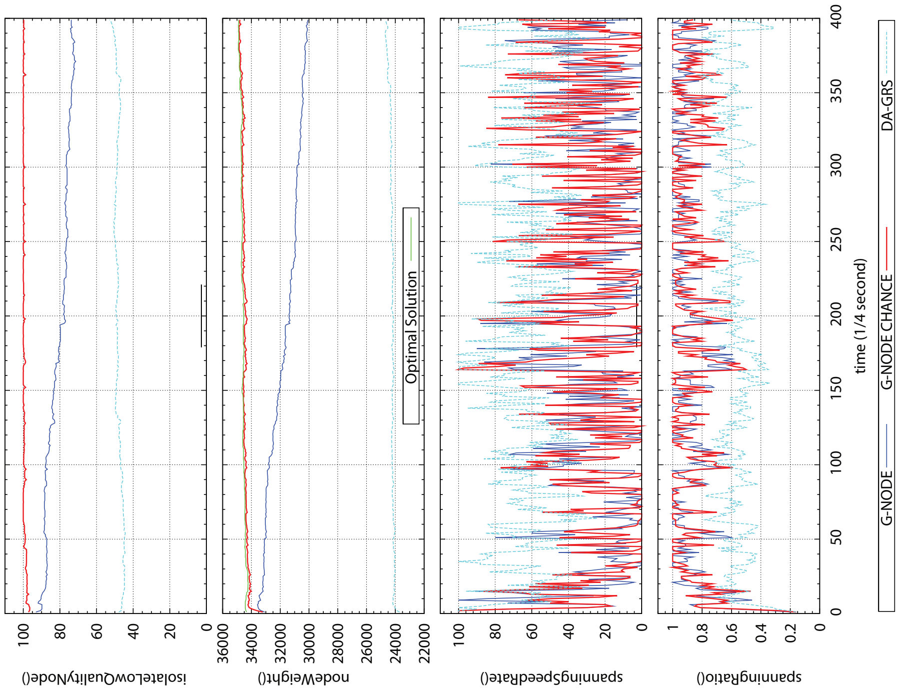


Figure A.23: G-NODE CHANCE on Shopping Mall Configuration 3

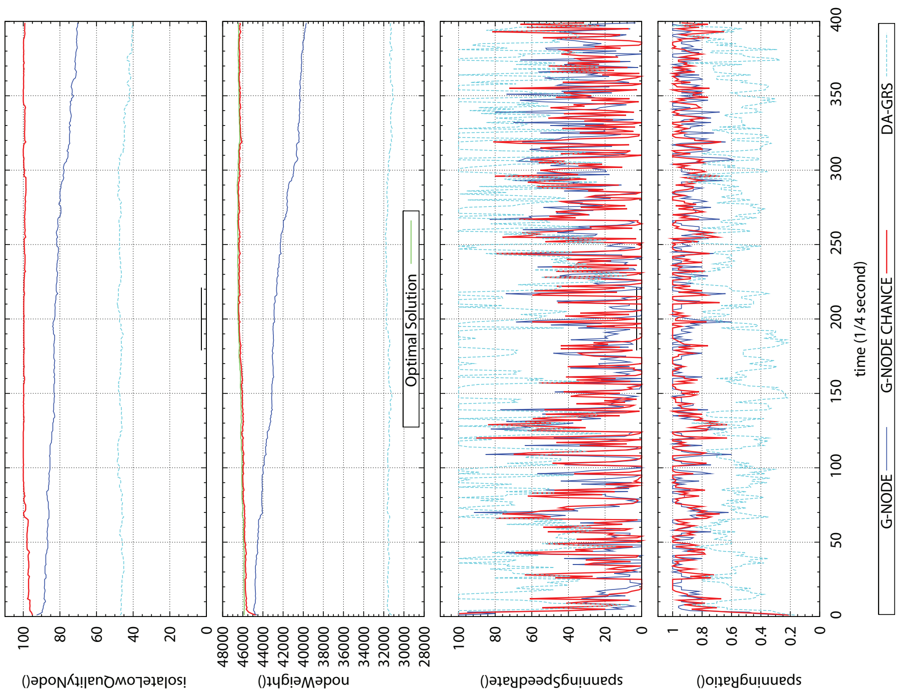


Figure A.24: G-NODE CHANCE on Shopping Mall Configuration 4

A.4 Overhead Messages for CHANCE heuristic

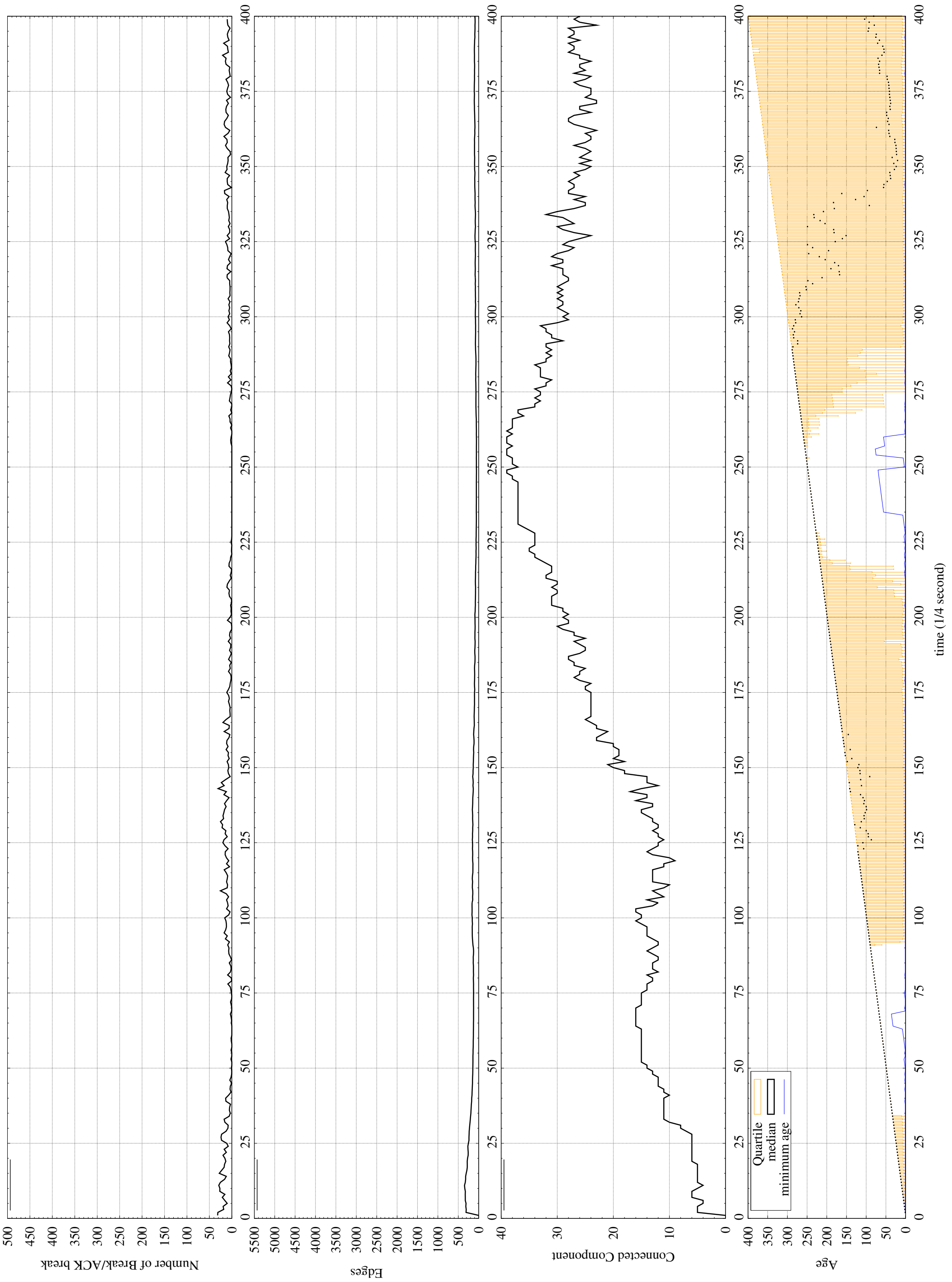


Figure A.25: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 1

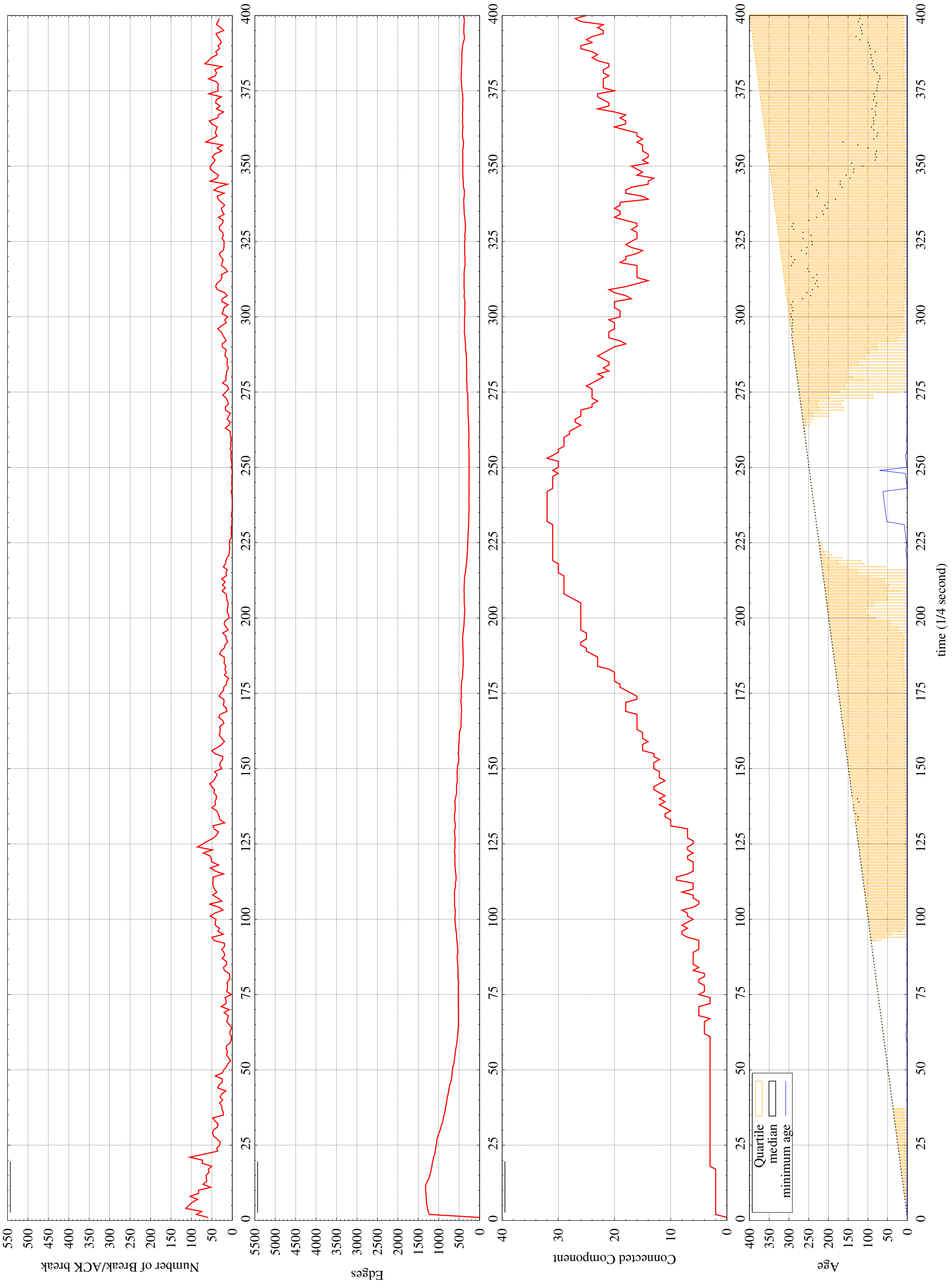


Figure A.26: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 2

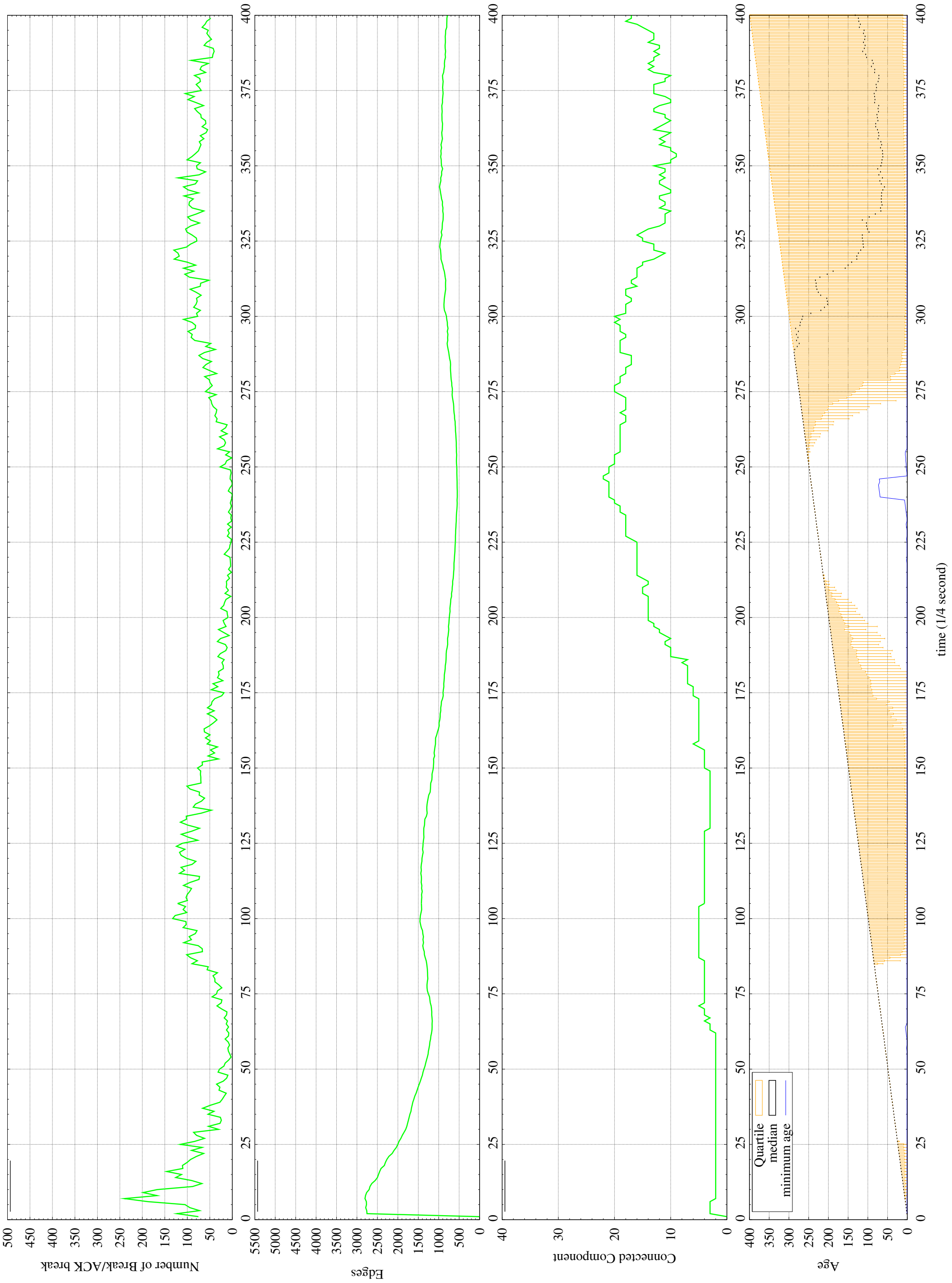


Figure A.27: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 3

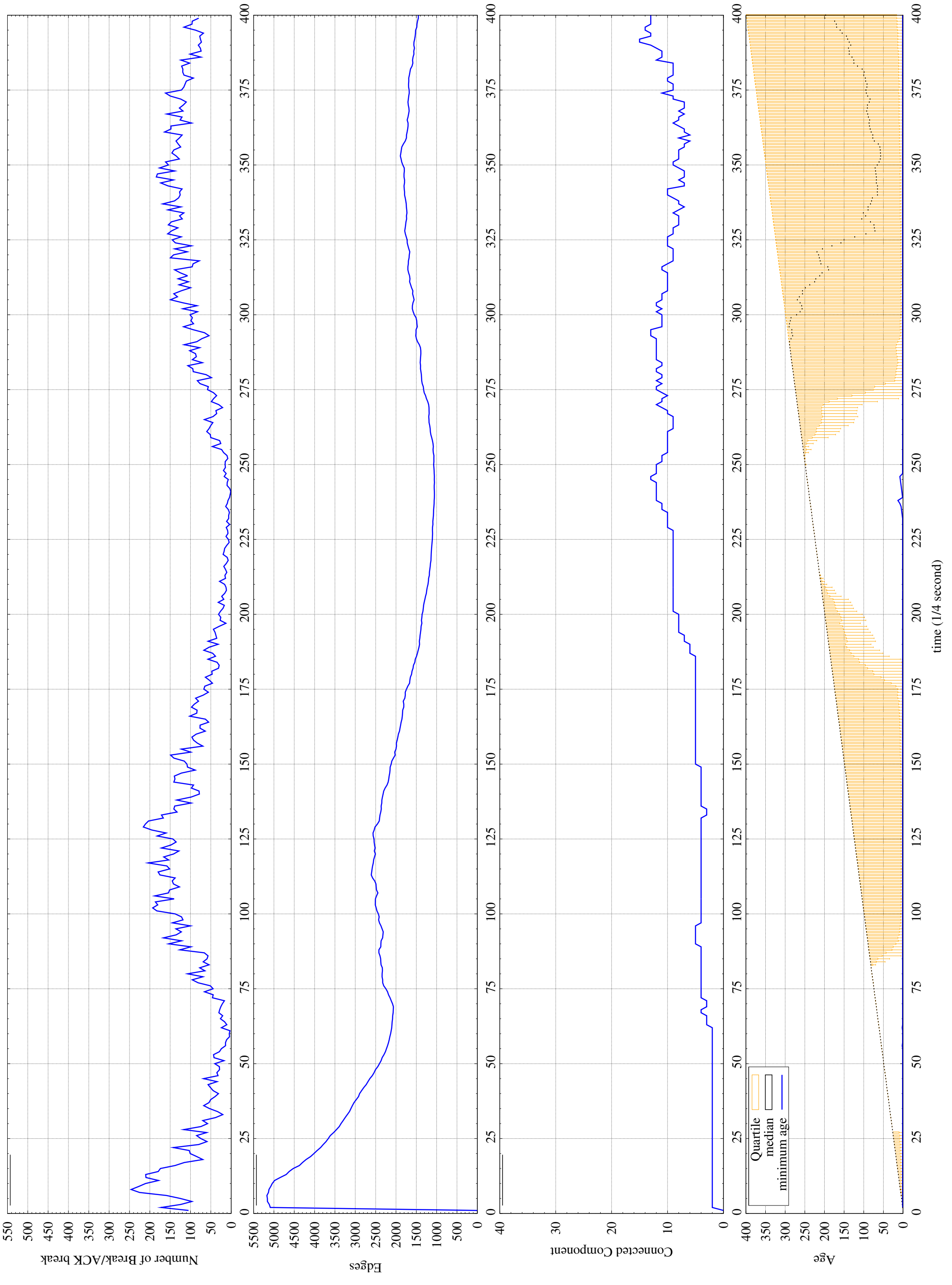


Figure A.28: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Highway Configuration 4

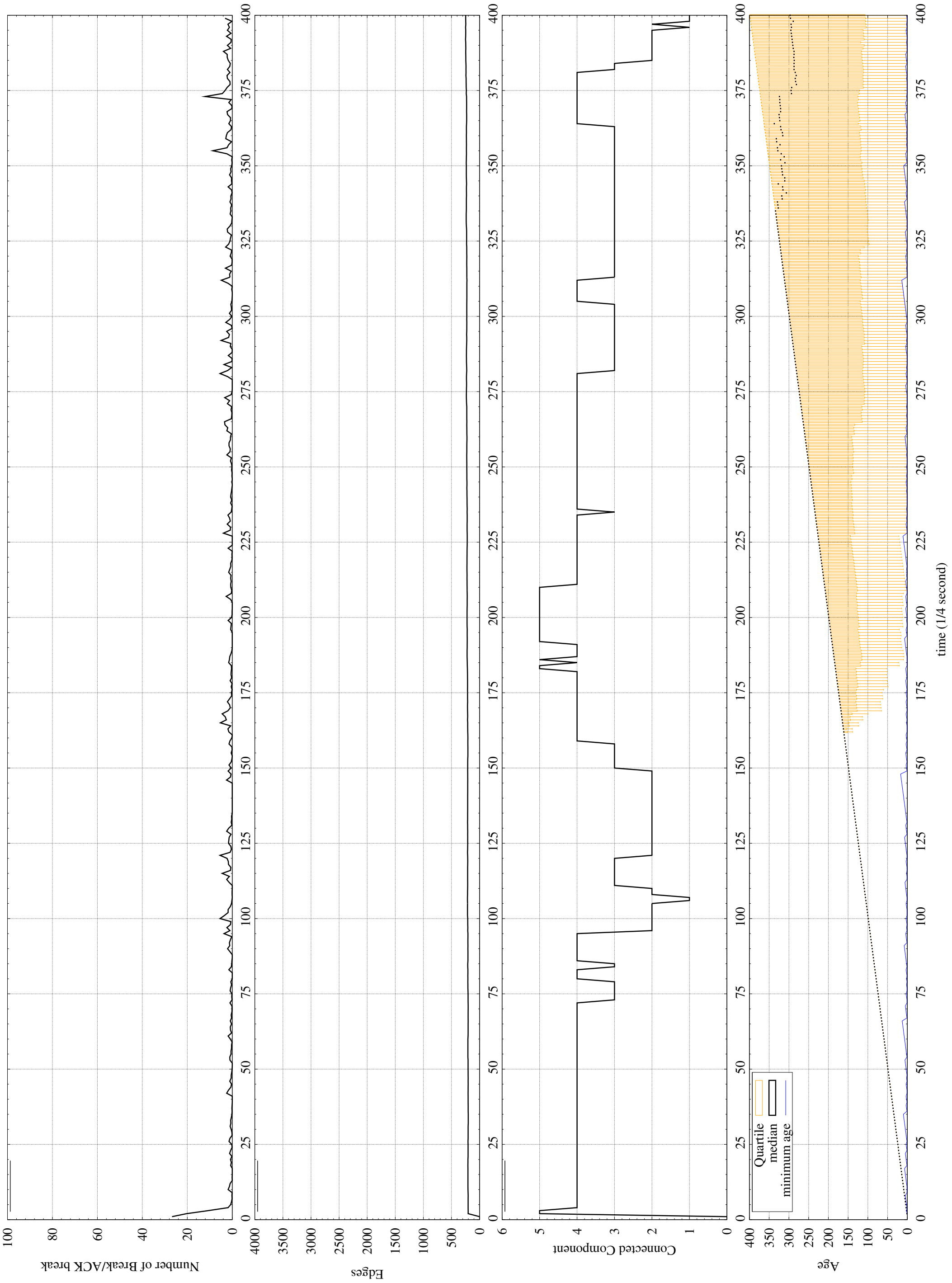


Figure A.29: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 1

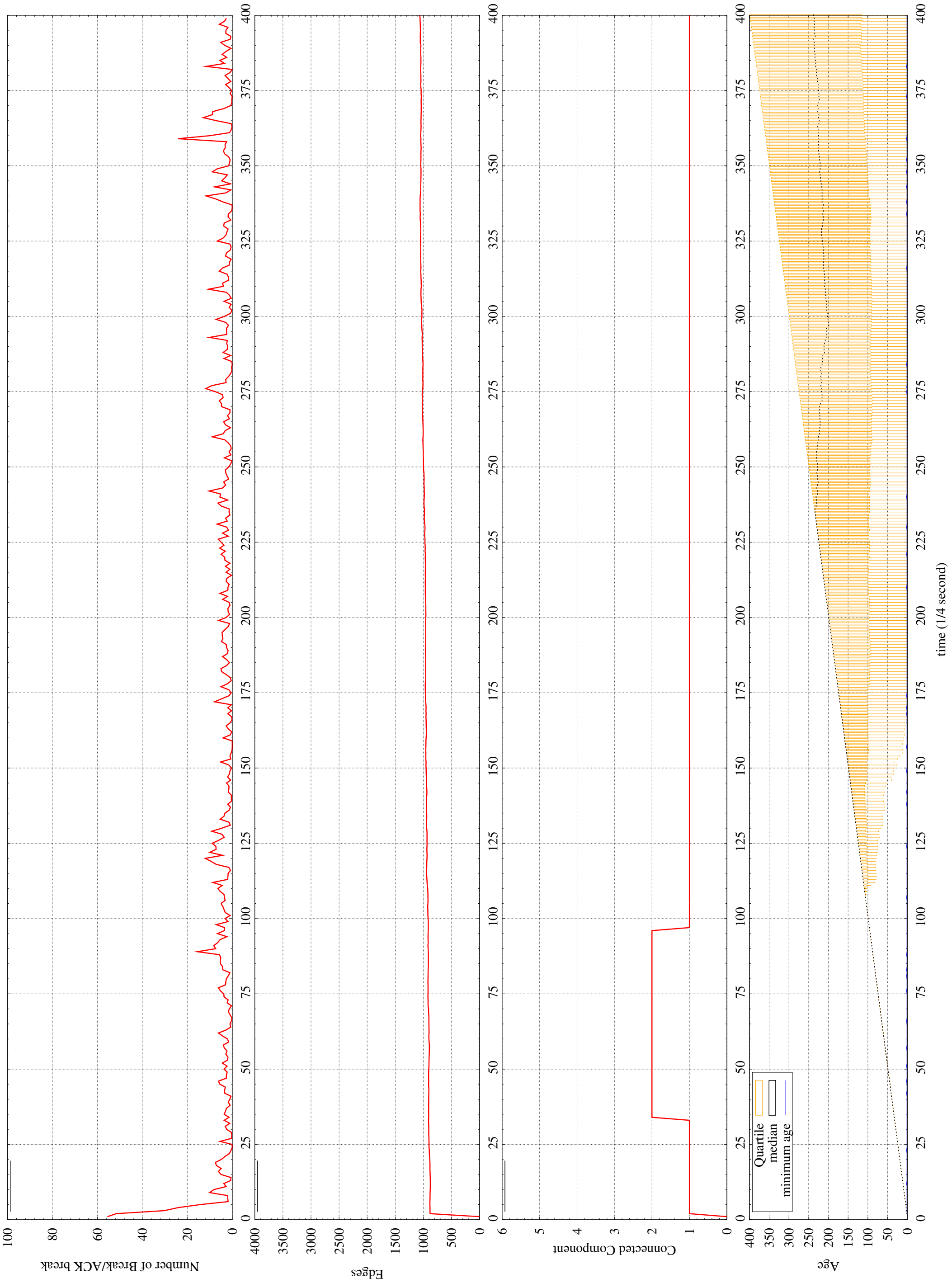


Figure A.30: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 2

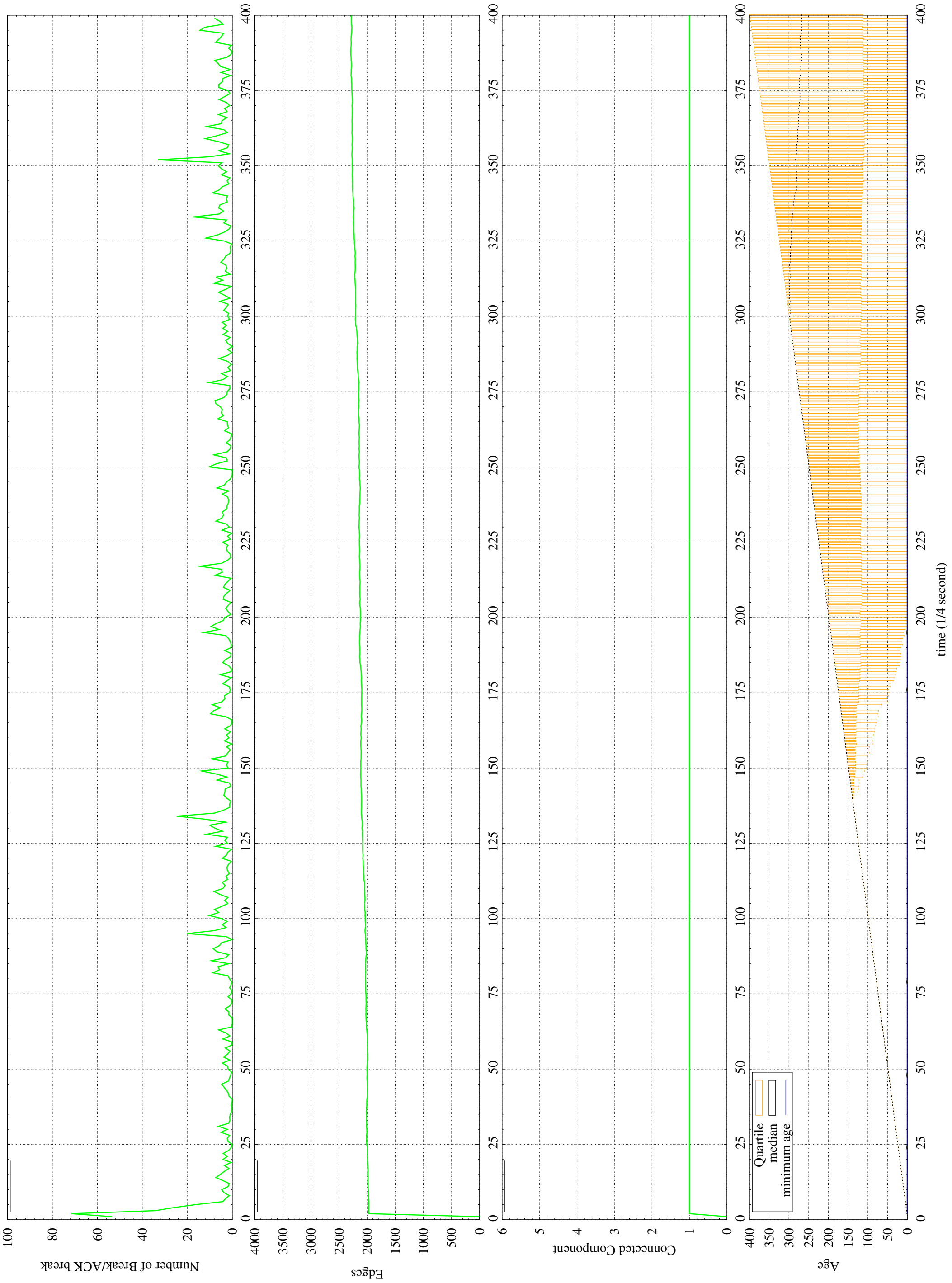


Figure A.31: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 3

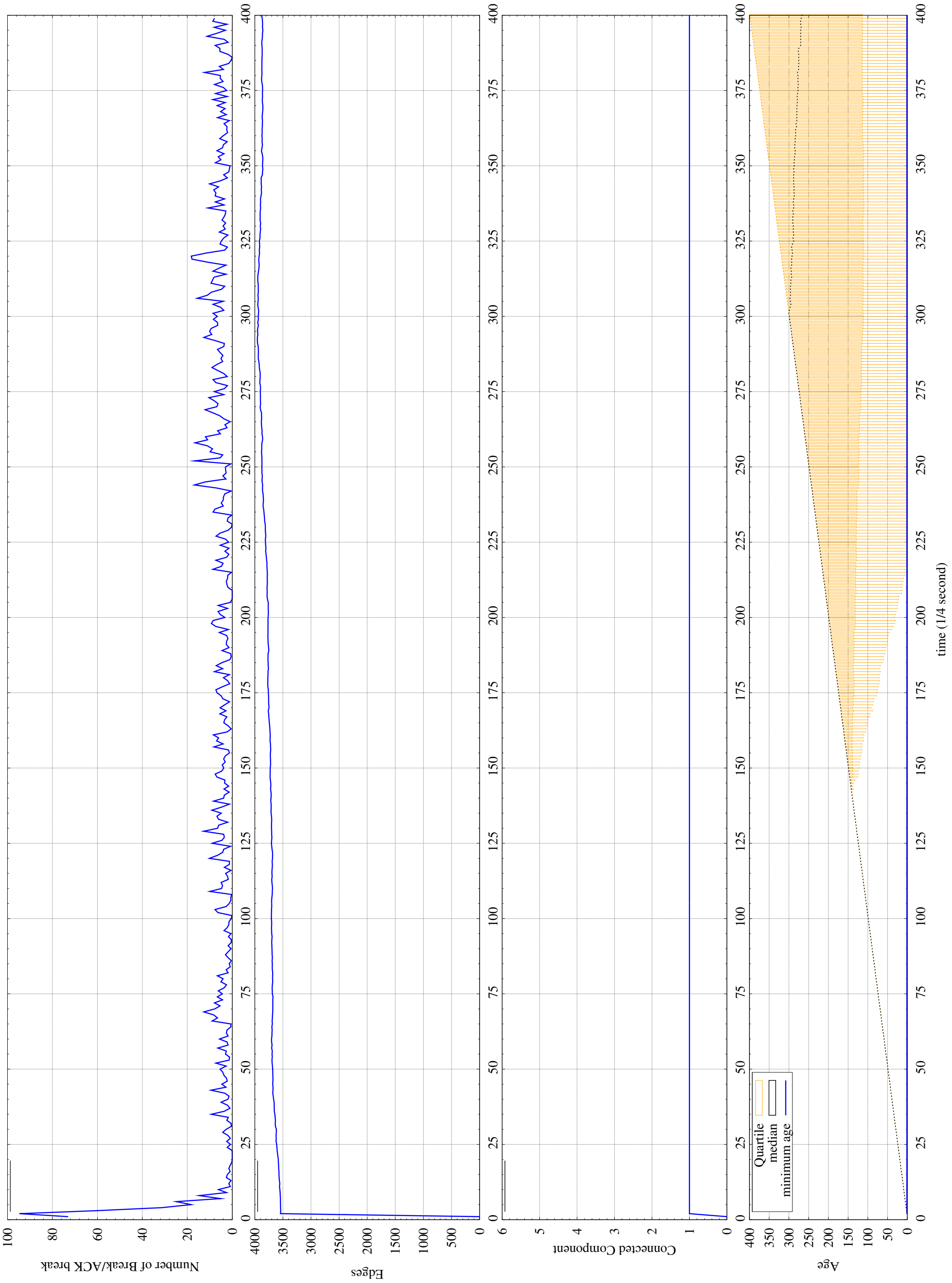


Figure A.32: Comparison of Overhead Messages used by CHANCE heuristic and Topology Characteristics on Shopping Mall Configuration 4

A.5 Hypervolume Result

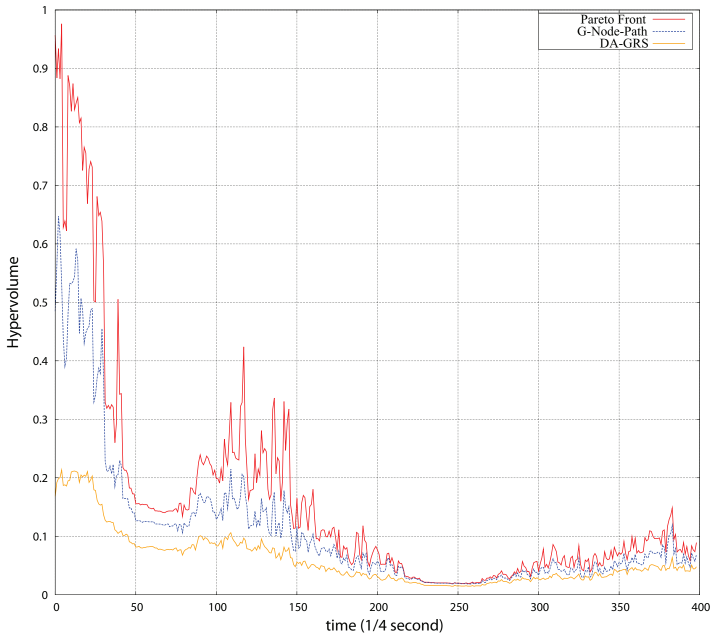


Figure A.33: Hypervolume: G-Node-Path, DA-GRS and Pareto front on Highway Configuration 1

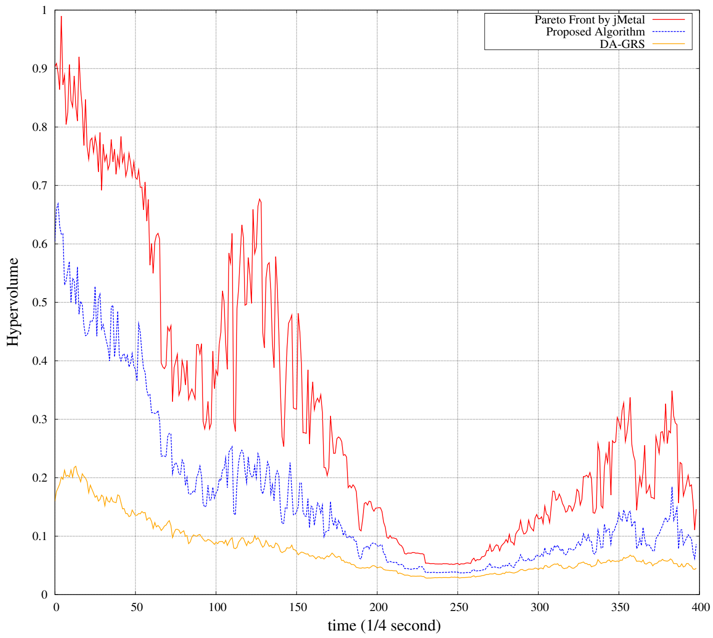


Figure A.34: Hypervolume: G-Node-Path, DA-GRS and Pareto front on Highway Configuration 2

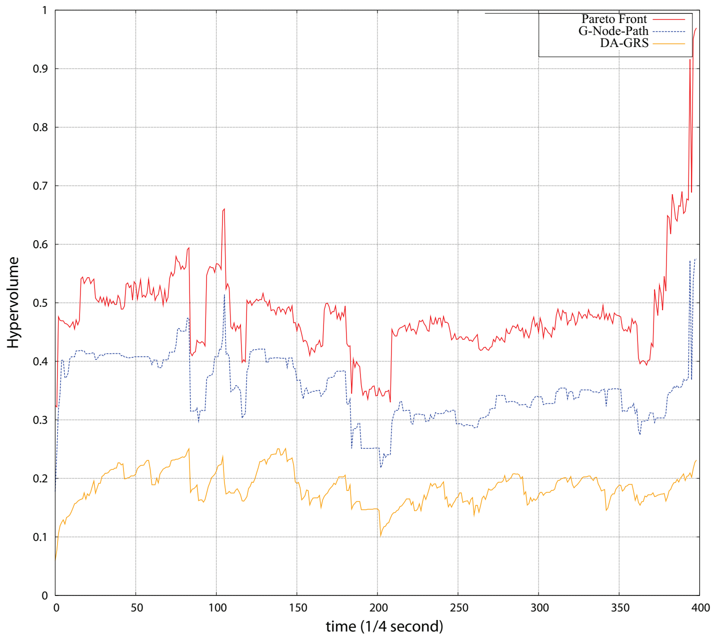


Figure A.35: Hypervolume: G-Node-Path, DA-GRS and Pareto front on shopping mall Configuration 1

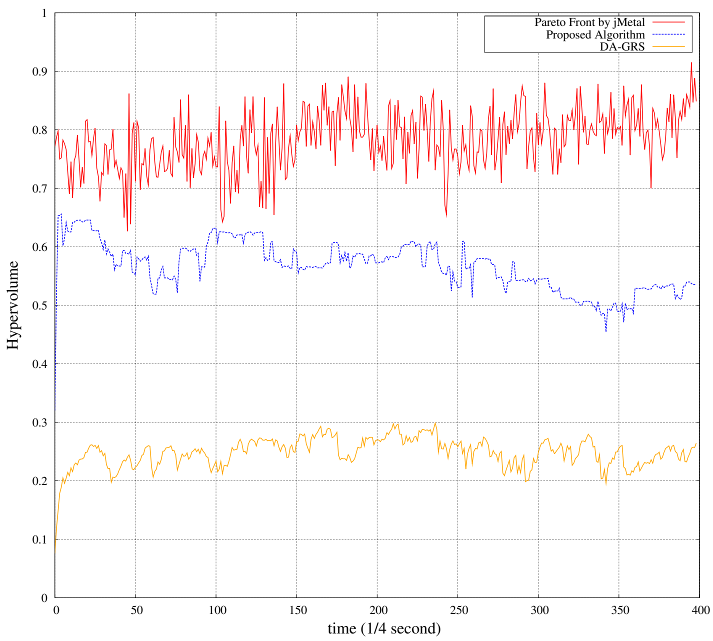


Figure A.36: Hypervolume: G-Node-Path, DA-GRS and Pareto front on shopping mall Configuration 2

A.6 Robustness Metric Result for Multi-objective problem

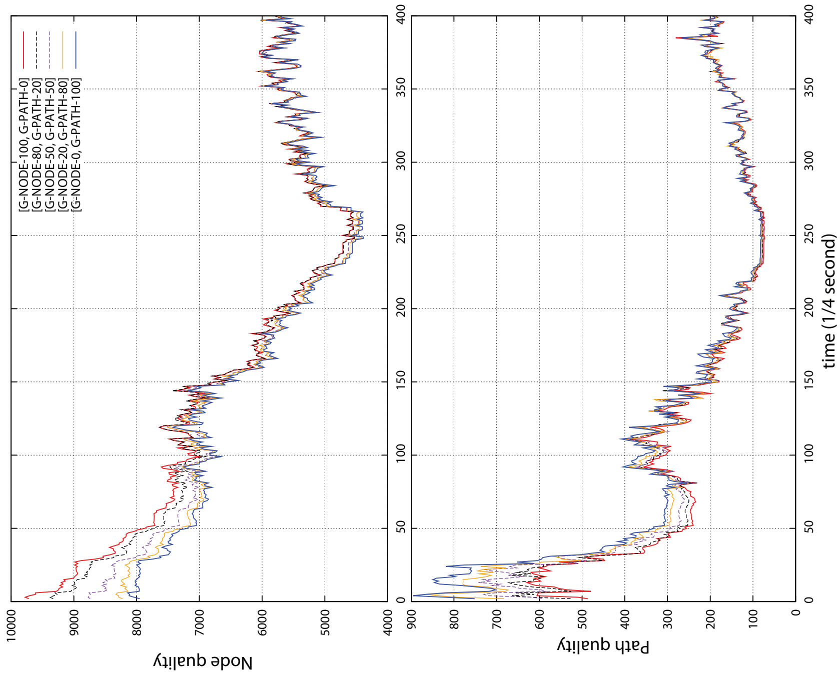


Figure A.37: G-Node-Path at different preference value on Highway Configuration 1

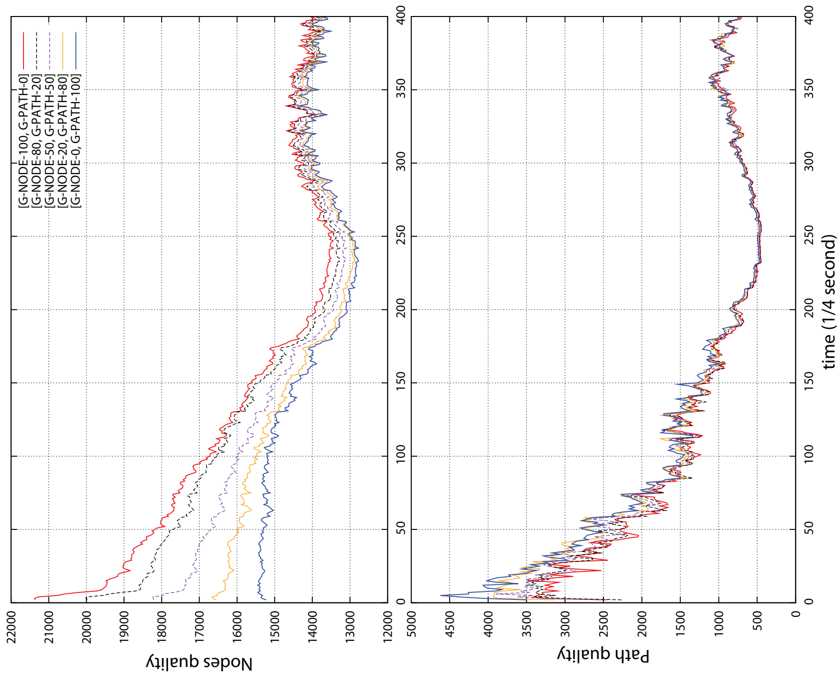


Figure A.38: G-Node-Path at different preference value on Highway Configuration 2

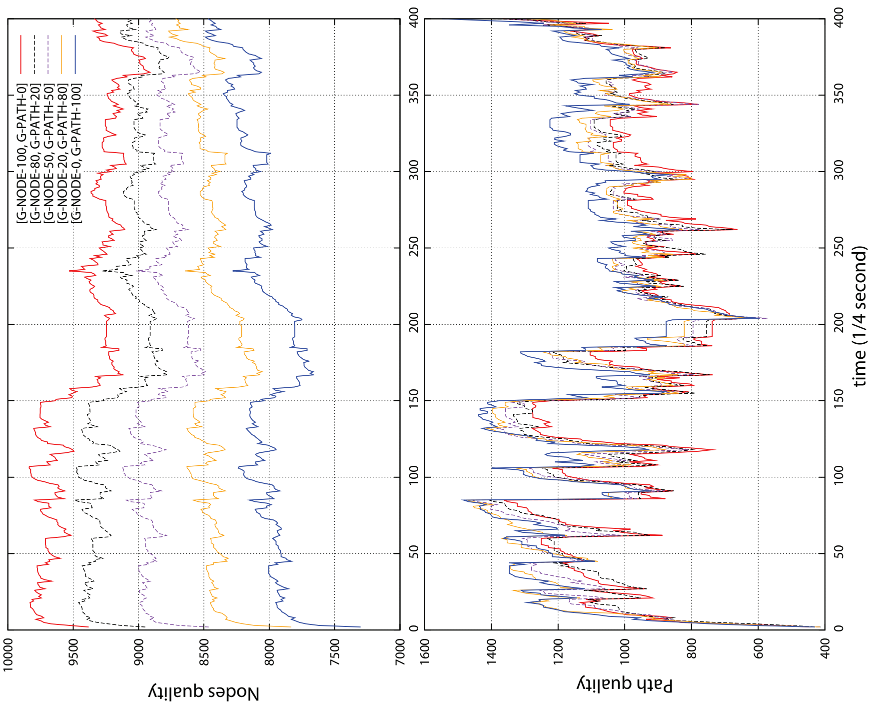


Figure A.39: G-Node-Path at different preference value on shopping mall Configuration 1

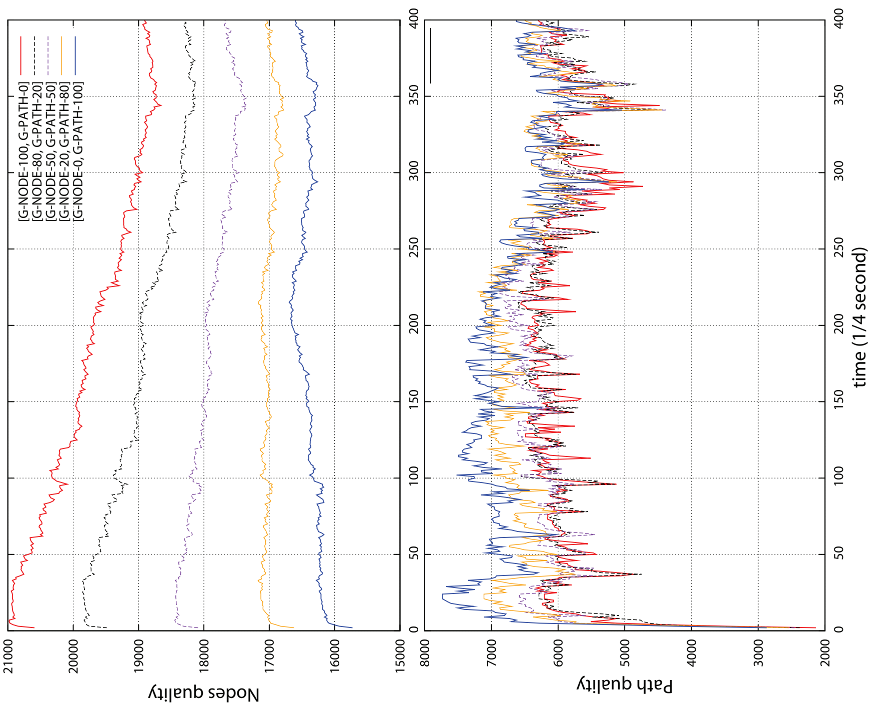


Figure A.40: G-Node-Path at different preference value on shopping mall Configuration 2

References

[1] E. Alba and B. Dorronsoro, *Cellular Genetic Algorithms.* Operations Research/Computer Science Interfaces, Springer Science+Business Media, LLC, 2008. viii, 27

[2] "Typical discharge discharge curves for cells using a range of cell chemistries when discharged at 0.2c rate." website: http://www.mpoweruk.com/performance.htm. viii, 66

[3] "Lithium ion polymer technical data." website: http://www.ibt-power.com/Battery$_{p}acks/Li_{P}olymer/Lithium_{p}olymer_{t}ech.html.$,

[4] TeliaSonera, "Teliasonera first in the world with 4g services." website, December 2009. last checked: June 1, 2010. 8

[5] T. 3rd Generation Partnership Project (3GPP) unites telecommunications standards bodies, "Lte." website (http://www.3gpp.org/LTE). last checked: June 1, 2010. 8

[6] H. Corporation, "HTC desire." website. last checked: May 17, 2010. 9

[7] I. Anand Lal Shimpi, AnandTech, "iphone 3gs performance: A significant performance bump." website, June 2009. last checked: May 17, 2010. 9

[8] PDAdb.net, "At&t blackberry bold 9700 (rim onyx) specs." website, November 2009. last checked: May 17, 2010. 9

[9] BlackBerry, Belgium & Luxembourg, "Blackberry bold 9700." website. last checked: May 17, 2010. 9

[10] I. Motorola, "Droid by motorola." website. last checked: May 17, 2010. 9

[11] MOTODEV: The Motorola develper network, "Droid™ By Motorola, A855." website. last checked: May 17, 2010. 9

[12] C. E. Perkins, *Ad hoc networking.* Addison-Wesley Longman Publishing Co., Inc., 2001. 11

[13] T. Clausen, "Manet neighborhood discovery protocol (nhdp), draft-ietf-manet-nhdp-03." website, May 2007. 11

[14] C. S. R. Murthy and B. Manoj., *Ad Hoc wireless networks: architectures and protocols.* Pearson Education, Inc., 2004. 12

[15] S. Corson and J. Macker, "Mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations." IETF.org, January 1999. 12, 13

[16] IBM Corporation, "Autonomic computing: Overview - the 8 elements." http://www.research.ibm.com/autonomic/overview/elements.html, 2001. last checked: June 2, 2010. 14

[17] D. Bein, *Guide to Wireless Ad Hoc Networks*, ch. 2, pp. 27–42. The Computer Communication and Networks, Springer-Verlag London Limited, 2009. 14

[18] "Car 2 car communication consortium." website. last checked: June 7. 15

[19] B. Karp and H. T. Kung, "Gpsr: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pp. 243–254, ACM, 2000. 16

[20] B. N. Karp, "Geographic routing for wireless networks," tech. rep., Harvard University, 2000. 16

[21] B. C. Seet, G. Liu, B. S. Lee, C. H. Foh, K. J. Wong, and K. K. Lee, "A-STAR: a mobile ad hoc routing strategy for metropolis vehicular communications," *Lecture Notes in Computer Science : NETWORKING 2004, Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communications*, pp. 989–999, 2004. 16

[22] A. Raniwala and T. cker Chiueh, "Architecture and algorithms for an ieee 802.11-based multi-channel wireless mesh network," in *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE*, vol. 3, pp. 2223–2234, IEEE, March 2005. 16

[23] I. F. AKYILDIZ and X. WANG, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, pp. S23 – S30, September 2005. 16

[24] "IEEE 802.11s Tutorial: Overview of the Amendment for Wireless Local Area Mesh Networking." website, November 2006. last checked: June 7, 2010. 17

[25] Wikipedia, "IEEE 802.11s." website. last checked: June 7, 2010. 17

[26] "IEEE P802.11 - task groups - meeting update." website. last checked: June 7, 2010. 17

[27] Delay Tolerant Networking Research Group, "Delay tolerant networking research group." website. last checked: June 7, 2010. 17

[28] V. Cerf, S. Burleigh, A. Hooke, L. Torgerson, R. Durst, K. Scott, K. Fall, and H. Weiss, "Delay-tolerant networking architecture (request for comments: 4838)." website, April 2007. last checked: June 7, 2010. 17

[29] P. N. Pathirana, N. Bulusu, A. V. Savkin, and S. Jha, "Node localization using mobile robots in delay-tolerant sensor networks," *IEEE Transactions on Mobile Computing*, vol. 4, pp. 285–296, May 2005. 18

[30] P. Juang, H. Oki, Y. Wang, M. Martonosi, L.-S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: Design tradeoffs and early experiences with zebranet," in *Proceedings of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS-X)*, vol. 37, pp. 96–107, ACM, October 2002. 18

[31] M. Abdulla and R. Simon, "A simulation analysis of multicasting in delay tolerant networks," in *WSC '06: Proceedings of the 38th conference on Winter simulation*, pp. 2234–2241, Winter Simulation Conference, 2006. 18

[32] W. Zhao and M. H. Ammer, "Message ferrying: Proactive routing in highly-partitioned wireless ad hoc networks," in *Proceedings of the The Ninth IEEE Workshop on Future Trends of Distributed Computing Systems*, p. 308, IEEE Computer Society, May 2003. 18

[33] W. Zhao, M. Ammar, and E. Zegura, "A message ferrying approach for data delivery in sparse mobile ad hoc networks," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*, pp. 187–198, ACM, May 2004. 18

[34] L. Pelusi, A. Passarella, and M. Conti, "Opportunistic networking: Data forwarding in disconnected mobile ad hoc networks," *IEEE Communications Magazine*, vol. 44, pp. 134–141, November 2006. 18

[35] M. Gunes and M. Wenig, "Models for realistic mobility and radio wave propagation for ad hoc network simulations," in *Guide to Wireless Ad Hoc Networks*, Computer Communications and Networks, ch. 11, pp. 255–280, Springer London, 2009. 20

[36] M. L. Sichitiu, "Mobility models for ad hoc networks," in *Guide to Wireless Ad Hoc Networks*, Computer Communications and Networks, ch. 10, pp. 237 – 254, Springer London, 2009. 22

[37] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*, pp. 153–181. Kluwer Academic Publishers, 1996. 22

[38] F. Bai, "Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for adhoc networks," in *IEEE INFOCOM*, pp. 825–835, 2003. 22

[39] M. Sanchez and P. Manzoni, "A java-based ad hoc networks simulator," in *the SCS Western Multiconference Web-based Simulation Track*, January 1999. 22

[40] L. Hogie, *Mobile Ad Hoc Networks: Modelling, Simulation and Broadcast-based Applications*. PhD thesis, University of Le Havre (France), University of Luxembourg (Luxembourg), April 2007. 22, 99

[41] C. H. Papadimitriou and K. Steiglitz, *Combinatorial optimization: algorithms and complexity*. Courier Dover Publications, 1998. 23

[42] E. K. P. Chong and S. H. Zak, *An introduction to optimization.* John Wiley & Sons, Ltd, 3rd ed., 2008. 23

[43] K. Deb, *Multi-Objective Optimization using Evolutionary Algorithms.* Wiley-Interscience Series in Systems and Optimization, John Wiley & Sons, Chichester, 2001. 23, 26, 28

[44] C. A. C. Coello and G. B. Lamont, eds., *Applications of Multi-Objective Evolutionary Algorithms.* Advances in Natural Computation - Vol. 1, World Scientific Publishing Co. Pet. Ltd., 2004. 23, 24

[45] C. H. Papadimitriou, *Computational Complexity.* Addison-Wesley Longman Publishing Co., Inc., 1994. 26

[46] J. B. Kruskal, "On the shortest spanning subtree of a graph and the traveling salesman problem," in *Proceedings of the American Mathematical Society*, vol. 7, pp. 48–50, February 1956. 26, 43

[47] R. C. Prim, "Shortest connection networks and some generalizations," *Bell System Technology Journal*, vol. 36, pp. 1389–1401, 1957. 26, 43

[48] G. L. Nemhauser and L. A. Wolsey, *Integer and combinatorial optimization.* Wiley-Interscience, 1988. 27

[49] C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: Overview and conceptual comparison," *ACM Comput. Surv.*, vol. 35, no. 3, pp. 268–308, 2003. 27

[50] C. C. Skiścim and B. L. Golden, "Optimization by simulated annealing: A preliminary computational study for the tsp," in *WSC '83: Proceedings of the 15th conference on Winter Simulation*, (Piscataway, NJ, USA), pp. 523–535, IEEE Computer Society Press, 1983. 27

[51] Glover and Laguna, *Tabu Search.* Kluwer Academic Publishers, 1997. 27

[52] M. Dorigo and T. Stützle, *Ant Colony Optimization.* Scituate, MA, USA: Bradford Company, 2004. 28

[53] J. K. Riccardo Poli and T. Blackwell, "Particle swarm optimization: An overview," *Swarm Intelligence*, vol. 1, pp. 33–57, August 2007. 28

[54] F. Glover, M. Laguna, and R. Martí, "Fundamentals of scatter search and path relinking," *Control and Cybernetics*, vol. 39, pp. 653–684, 2000. 28

[55] P. Santi, *Topology control in wireless ad hoc and sensor networks.* West Sussex, England: John Wiley & Sons, Ltd, 2005. 33, 37

[56] M. R. Brust and S. Rothkugel, "A taxonomic approach to topology control in ad hoc and wireless networks," in *ICN '07: Proceedings of the Sixth International Conference on Networking*, (Washington, DC, USA), p. 25, IEEE Computer Society, 2007. 33

[57] D. Tian and N. D. Georganas, "A coverage-preserving node scheduling scheme for large wireless sensor networks," in *WSNA '02: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, (New York, NY, USA), pp. 32–41, ACM, 2002. 33

[58] M. Augusto, M. Vieira, L. Filipe, M. Vieira, L. B. Ruiz, A. A. F. Loureiro, A. O. Fern, J. Marcos, and S. Nogueira, "Scheduling nodes in wireless sensor networks: A voronoi approach," in *28th Annual IEEE International Conference on Local Computer Networks (LCN'03)*, pp. 423–429, October 2003. 33

[59] R. Rajaraman, "Topology control and routing in ad hoc networks: A survey," *SIGACT News*, vol. 33, pp. 60–73, 2002. 34

[60] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar, "Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the compow protocol," in *in European Wireless Conference*, pp. 152–162, 2002. 34

[61] D. M. Blough, M. Leoncini, G. Resta, and P. Santi, "The k-neigh protocol for symmetric topology control in ad hoc networks," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, (New York, NY, USA), pp. 141–152, ACM, 2003. 34

[62] L. J. Cowen, "Compact routing with minimum stretch," in *SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, (Philadelphia, PA, USA), pp. 255–260, Society for Industrial and Applied Mathematics, 1999. 34

[63] R. Flury, S. V. Pemmaraju, and R. Wattenhofer, "Greedy routing with bounded stretch," in IEEE *INFOCOM 2009*, pp. 1737–1745, 2009. 34

[64] J.-H. Lin, C.-R. Dow, and S.-F. Hwang, "A distributed virtual backbone development scheme for ad-hoc wireless networks," *Wirel. Pers. Commun.*, vol. 27, no. 3, pp. 215–233, 2003. 34

[65] B. Liang and Z. J. Haas, "Virtual backbone generation and maintenance in ad hoc network mobility management," in *Proceedings of IEEE INFOCOM*, pp. 1293–1302, 2000. 34

[66] T. W. Haynes, S. Hedetniemi, and P. Slater, *Fundamentals of Domination in Graphs.* Marcel Dekker, Inc., 1998. 35

[67] F. V. Fomin, D. Kratsch, J. Woeginger, and G. J. Woeginger, "Exact (exponential) algorithms for the dominating set problem," in *Graph Theoretic Concepts in Computer Science*, vol. Volume 3353/2005 of *Lecture Nodes in Computer Science*, pp. 245–256, Springer Berline/ Heidelberg, 2004. 35

[68] I. Stojmenovic, M. Seddigh, and J. Zunic, "Dominating sets and neighbor elimination-based broadcasting algorithms in wireless networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, pp. 14–25, 2002. 35

[69] S. Butenko, X. Cheng, D. zhu Du, and P. M. Pardalos, "On the construction of virtual backbone for ad hoc wireless network," in *Cooperative Control: Models, Applications and Algorithms*, pp. 43–54, Kluwer Academic Publishers, 2003. 36

[70] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*, pp. 7–14, ACM, 1999. 36

[71] F. Dai and J. Wu, "An extended localized algorithm for connected dominating set formation in ad hoc wireless networks," *IEEE Transactions On Parallel and Distributed Systems*, vol. 15, pp. 908–920, October 2004. 36

[72] J. Schleich, P. Bouvry, and L. T. H. An, "Decentralized fault-tolerant connected dominating set algorithm for mobile ad hoc networks," in *ICWN* (H. R. Arabnia and V. A. Clincy, eds.), pp. 354 – 360, CSREA Press, 2009. 36, 53

[73] H. Luo, F. Ye, J. Cheng, S. Lu, and L. Zhang, "Ttdd: Two-tier data dissemination in large-scale wireless sensor networks," *Wireless Networks*, vol. 11, pp. 161–175, January 2005. 37

[74] A. Ephremides, J. Wieselthier, and D. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 56–73, 1987. 37

[75] M. Chatterjee, S. K. Das, and D. Turgut, "Wca: A weighted clustering algorithm for mobile ad hoc networks," *Cluster Computing*, vol. 5, pp. 193–204, April 2004. 37

[76] M. R. Brust, A. Andronache, and S. Rothkugel, "Waca: A hierarchical weighted clustering algorithm optimized for mobile hybrid networks," *CoRR*, 2007. 37

[77] F. Li, Y. Yang, and J. Wu, "Mobility management in manets: Exploit the positive impacts of mobility," in *Guide to Wireless Ad Hoc Networks*, Computer Communications and Networks, ch. 9, pp. 211–235, Springer London, 2009. 37

[78] M. V. Marathe, R. Ravi, R. Sundaram, S. S. Ravi, D. J. Rosenkrantz, and H. B. Hunt, III, "Bicriteria network design problems," *Journal of Algorithms*, vol. 28, pp. 142–171, July 1998. 40

[79] "standard 802.1D - MAC bridges." IEEE Computer Society, 1993. 40, 44

[80] R. Gandhi, A. Mishra, and S. Parthasarathy, "Minimizing broadcast latency and redundancy in ad hoc networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 4, pp. 840–851, 2008. 40, 58

[81] P. Ruiz, B. Dorronsoro, D. Khadraoui, and P. Bouvry, "BODYF–A Parameterless Broadcasting Protocol Over Dynamic Forest," in *Workshop on Optimization Issues in Grid and Parallel Computing Environments, part of the High Performance Computing and Simulation Conference (HPCS)*, pp. 297–303, 2008. 40, 60

[82] Jüttner, Alpár and Magi, Ádám, "Tree based broadcast in ad hoc networks," *Mobile Networking Applications*, vol. 10, no. 5, pp. 753–762, 2005. 40, 41

[83] S. Srivastava and R. K. Ghosh, "Cluster based routing using a k-tree core backbone for mobile ad hoc networks," in *DIALM '02: Proceedings of the 6th international workshop on Discrete algorithms and methods for mobile computing and communications*, (New York, NY, USA), pp. 14–23, ACM, 2002. 40

[84] M. Marathe, H. Breu, H. B. H. Iii, S. S. Ravi, and D. J. Rosenkrantz, "Simple heuristics for unit disk graphs," *Networks*, vol. 25, pp. 59–68, 1995. 40

[85] K. M. Alzoubi, P. jun Wan, and O. Frieder, "Distributed heuristics for connected dominating sets in wireless ad hoc networks," *Journal of Communications and Networks*, vol. 4, pp. 22–29, 2002. 40

[86] V. V. Vazirani, *Approximation Algorithms*. Springer-Verlag Berlin Heidelberg, 2001. 40

[87] F. C. Gartner, "A survey of self-stabilizing spanning-tree construction algorithms," tech. rep., 2003. 40

[88] S. Radhakrishnan, G. Racherla, C. N. Sekharan, N. S. V. Rao, and S. G. Batsell, "Protocol for dynamic ad-hoc networks using distributed spanning trees," *Wireless Networks*, vol. 9, no. 6, pp. 673–686, 2003. 41, 43

[89] A. Casteigts, "Model driven capabilities of the DA-GRS model," in *ICAS '06: Proceedings of the International Conference on Autonomic and Autonomous Systems*, p. 24, 2006. 41

[90] E. Sopena and Y. Metivier, "Graph relabeling systems: a general overview," *Computers and Artificial Intelligence*, vol. 16, no. 2, pp. 167–185, 1997. 41

[91] Y. Pigné, *Modélisation et Traitement Décentralisé des Graphes Dynamiques - Application Aux Réseaux Mobiles Ad Hoc.* Phd thesis, L'Université du Harve, December 2008. 43, 53, 58, 59, 70, 114

[92] O. Borůvka, "O jistém problému minimálmím.," *Práce Mor. Přírodověd. v Brně (Acta Societ. Scient. Natur. Moravicae)*, vol. 3, pp. 37–58, 1926. 43

[93] V. Jarník, "O jistém problému minimálmím. [about a certain minimal problem] (in czech)," *Práce Moravské Přírodovědecké Společnosti*, vol. 6, pp. 57–63, 1930. 43

[94] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd ed., 2001. 44

[95] R. L. Graham and P. Hell, "On the history of the minimum spanning tree problem," *IEEE Ann. Hist. Comput.*, vol. 7, no. 1, pp. 43–57, 1985. 44

[96] D. R. Cheriton and R. E. Tarjan, "Finding minimum spanning trees," *SIAM J. Comput.*, vol. 5, no. 4, pp. 724–742, 1976. 44

[97] R. G. Gallager, P. A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Trans. Program. Lang. Syst.*, vol. 5, no. 1, pp. 66–77, 1983. 44

[98] M. Faloutsos and M. Molle, "Optimal distributed algorithm for minimum spanning trees revisited," in *PODC '95: Proceedings of the fourteenth annual ACM symposium on Principles of distributed computing*, (New York, NY, USA), pp. 231–237, ACM, 1995. 44

[99] Z. Lotker, E. Pavlov, B. Patt-Shamir, and D. Peleg, "Mst construction in o(log log n) communication rounds," in *SPAA '03: Proceedings of the fifteenth annual ACM symposium on Parallel algorithms and architectures*, (New York, NY, USA), pp. 94–100, ACM, 2003. 44

[100] G. Chen, S. Chen, W. Guo, and H. Chen, "The multi-criteria minimum spanning tree problem based genetic algorithm," *Inf. Sci.*, vol. 177, no. 22, pp. 5050–5063, 2007. 45

[101] S. Ali, A. A. Maciejewski, H. J. Siegel, and J.-K. Kim, "Measuring the robustness of a resource allocation," *IEEE Transactions on Parallel and Distributed Systems*, vol. 15, no. 7, pp. 630–641, 2004. 46

[102] M. Gast, *802.11 Wireless Networks: The Definitive Guide.* USA: O'REILLY Media, Inc, 2nd ed., April 2005. 54

[103] "IEEE standard 802.11: Wireless lan medium access control and physical layer specifications." IEEE Computer Society, August 1999. 55, 109, 139

[104] L. Cardelli, "An implementation model of rendezvous communication." Lecture notes in Computer Science 197, 1985. 55

[105] Y. Pigné, A. Casteigts, F. Guinand, and S. Chaumette, "Construction et maintien d'une forêt couvrante dans un réseau dynamique," in *12èmes Rencontres Francophones sur les Aspects Algorithmiques de Télécommunications (AlgoTel)* (Maria Gradinariu Potop-Butucaru et Hervé Rivano, ed.), p. 4, 2010. 57

[106] N. Santoro and R. Khatib, "Labelling and implicit routing in networks," *Comput. J.*, vol. 28, no. 1, pp. 5–8, 1985. 58

[107] K. Raymond, "A tree-based algorithm for distributed mutual exclusion," *ACM Trans. Comput. Syst.*, vol. 7, no. 1, pp. 61–77, 1989. 58

[108] N. Malpani, Y. Chen, and J. L. Welch, "Distributed token circulation in mobile ad hoc networks," IEEE Transactions on Mobile Computing, vol. 4, no. 2, pp. 154–165, 2005. 58

[109] E. G. Goodaire and M. M. Parmenter, *Discrete mathematics with graph theory.* Printice-Hall, Inc., 2nd ed., 2002. 58

[110] N. Bauer, M. Colagrosso, and T. Camp, "An agile approach to distributed information dissemination in mobile ad hoc networks," in *IEEE International Symposium on a*

World of Wireless, Mobile and Multimedia Networks (WoWMoM), pp. 131–141, 2005. 60

[111] I. Stojmenovic, M. Russell, and B. Vukojevic, "Depth first search and location based localized routing and qos routing in wireless networks," in *Intl. Conf. on Parallel Processing (ICPP'00)*, 2000. 60

[112] G. F. Marias, P. Georgiadis, D. Flitzanis, and K. Mandalas, "Cooperation enforcement schemes for manets: a survey," *Wirel. Commun. Mob. Comput.*, vol. 6, no. 3, pp. 319–332, 2006. 62, 63

[113] S. Buchegger and J.-Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems," *IEEE Communications Magazine*, vol. 43, pp. 101–107, July 2005. 62

[114] S. Marsh, *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994. 62

[115] L. Mui, M. Mohtashemi, and A. Halberstadt, "Notions of reputation in multiagents systems: a review," in *AAMAS '02: Proceedings of the first international joint conference on Autonomous agents and multiagent systems*, (New York, NY, USA), pp. 280–287, ACM, 2002. 62

[116] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "An integrated trust and reputation model for open multi-agent systems," *Autonomous Agents and Multi-Agent Systems*, vol. 13, no. 2, pp. 119–154, 2006. 62, 63

[117] M. Seredyński, *An Evolutionary Approach Towards Cooperation Enforcement in Ad Hoc networks*. PhD thesis, University of Luxembourg (Luxembourg), January 2009. 63

[118] S. Marti, T. J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, (New York, NY, USA), pp. 255–265, ACM, 2000. 63

[119] K. Wrona, "Distributed security: Ad hoc networks & beyond," *presented at the Ad Hoc Network Security Pampas Workshop, RHUL, London, September 16–17, 2002.*, 2002. 63

[120] J. V. D. Merwe, D. Dawoud, and S. McDonald, "A survey on peer-to-peer key management for mobile ad hoc networks," *ACM Comput. Surv.*, vol. 39, no. 1, p. 1, 2007. 63

[121] J. Golson, "iPhone 3GS Teardown Reveals All, Sort Of." http://gigaom.com/2009/06/19/iphone-3gs-teardown-reveals-all-sort-of/, June 2009. last checked: 24 August 2010. 67

[122] NexTag Comparison Shopping, "Nokia lithium ion cell phone battery - lithium ion (li-ion)." http://www.nextag.com/Nokia-Battery-3032639/specs-html. last checked: 24 August 2010. 67

[123] Dell, "Dell latitude d620 product details." http://preview.tinyurl.com/yeu9bw4. last checked: 24 August 2010. 67

[124] www.apple.com, "MacBook Pro: Tech Specs." http://www.apple.com/macbookpro/specs.html. 67

[125] R. Dube, C. D. Rais, K. yeh Wang, and S. K. Tripathi, "Signal stability based adaptive routing (ssa) for ad-hoc mobile networks," *IEEE Personal Communications*, vol. 4, pp. 36–45, 1997. 69

[126] J. J. Durillo, A. J. Nebro, F. Luna, B. Dorronsoro, and E. Alba, "jMetal: A Java Framework for Developing Multi-Objective Optimization Metaheuristics," Tech. Rep. ITI-2006-10, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga, E.T.S.I. Informática, Campus de Teatinos, December 2006. 86, 94

[127] A. Cayley, "A theorem on trees," *Quarterly Journal of Mathematics*, vol. 23, pp. 376–378, 1989. 92, 94

[128] G. Kirchoff, "über die auflosung der gleichungen, auf welche man dei der untersuchung der linearen verteilung galvanischer strome gefuhrt wird," *Ann. Phys. chem.*, vol. 72, pp. 497–508, 1847. 92

[129] R. J. Gould, *Graph Theory*. Benjamin-Cummings Pub Co, 1988. 92

[130] A. Shioura, A. Tamura, and T. Uno, "An optimal algorithm for scanning all spanning trees of undirected graphs," *SIAM J. Comput.*, vol. 26, no. 3, pp. 678–692, 1997. 92, 93

[131] L. Davis, D. Orvosh, A. Cox, and Y. Qiu, "A genetic algorithm for survivable network design," in *Proceedings of the 5th International Conference on Genetic Algorithms*, (San Francisco, CA, USA), pp. 408–415, Morgan Kaufmann Publishers Inc., 1993. 94

[132] S. Picciotto, *How to encode a tree*. PhD thesis, University of California, 1999. 94

[133] T. Paulden and D. K. Smith, "Some novel locality results for the blob code spanning tree representation," in *GECCO '07: Proceedings of the 9th annual conference on Genetic and evolutionary computation*, (New York, NY, USA), pp. 1320–1327, ACM, 2007. 94

[134] F. Rothlauf, *Representations for Genetic and Evolutionary Algorithms*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. 94

[135] G. Raidl and B. Julstrom, "Edge sets: an effective evolutionary coding of spanning trees," *Evolutionary Computation, IEEE Transactions on*, vol. 7, pp. 225–239, 2003. 94, 136

[136] L. Hogie, P. Bouvry, F. Guinand, G. Danoy, and E. Alba, "Simulating realistic mobility models for large heterogeneous MANETs," in *Demo proceeding of the 9th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM'06)*, p. 1, IEEE, October 2006. 99

[137] A. Dutot, F. Guinand, D. Olivier, and Y. Pigné, "Graphstream: A tool for bridging the gap between complex systems and dynamic graphs," in *EPNACS: Emergent Properties in Natural and Artificial Complex Systems*, October 2007. 99

[138] P. Gupta and P. R. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000. 108

[139] A. J. Nebro, F. Luna, E. Alba, B. Dorronsorro, J. J. Druillo, and A. Beham., "Abyss: Adapting scatter search to multiobjective optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, pp. 439–457, August 2008. 136

Publications

[SMCia08] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, “Trusted spanning tree for delay tolerant mobile ad hoc networks,” in *SMCia’08: IEEE Conference on Soft Computing in Industrial Applications*, pp. 131–136, 2008. 71

[EUC08] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, “Trusted spanning tree for delay tolerant MANETs,” in *2008 IEEE/IPIP International Conference on Embedded and Ubiquitous Computing (EUC 2008), Shanghai, China, December 17-20, 2008, Volume II: Workshops* (C.-Z. Xu and M. Guo, eds.), pp. 293–299, IEEE Computer Society, 2008. 79, 127

[IAIT09] A. Piyatumrong, P. Ruiz, P. Bouvry, F. Guinand, and K. Lavangnananda, “Token traversal strategies of a distributed spanning forest algorithm in mobile ad hoc - delay tolerant networks,” in *Advances in Information Technology*, vol. 55 of *Communications in Computer and Information Science*, pp. 96–109, Springer-Verlag Berlin Heidelberg, 2009. 54

[ICUMT09] A. Piyatumrong, P. Bouvry, F. Guinand, and K. Lavangnananda, “A study of token traversal strategies on tree-based backbones for mobile ad hoc - delay tolerant networks,” in *ICUMT’09: International Conference on Ultra Modern Telecommunications & Workshops*, pp. 1–8, 2009. 115

Index

A
Access Point (AP) 9
Ad hoc networks 11
Ad hoc wireless network 12
Approximated Methods 27
Autonomous system 13

B
Base station (BS) 8

C
Capacity Bandwidth (CB) 69
Classical Computer Networks 8
Computational Complexity 27
Conflicting Objectives 48
Cutting off power 33

D
DA-GRS 41
DA-GRS model 43
DA-GRS reference algorithm 43
DT-MANETs 18
DTNs 17
Dynamic Voltage Scaling (DVS) 33

E
Exact Methods 27

F
Flat Network Topology 37
Fully Connected Topology 32

G
Graph 20

H
Heuristics 27
Hierarchical Network Topology 35

I
IETF RFC2501 Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations 12
IETF RFC4838 DTNs Architectures 17

M
managed topology graph T 33
Managed Topology Subgraph 34
Meta-heuristics 27
Mobile Ad hoc Networks (MANETs) 13

O
Optimization 23

P
Pareto front 24
Partitioned Topology 32
Path 69

S
Self-Organization 13
Spanning Forest 40
Spanning Tree 40

T
Topology 30
Topology Control 33
Topology Management 32
Traced file 22
Transmission Range 33
tree_degree 71

W
WSNs 14