Improving Signalling Performance of Proactive MANET Routing Protocols

Yangcheng Huang

A dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy of the
University of London.

Department of Computer Science
University College London

2007
I, [Yangcheng Huang], confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.
Abstract

A mobile ad hoc network (MANET) comprises a set of nodes connected by radio wireless links in a temporary manner. The topology of a MANET may change rapidly and unpredictably because of node mobility. Resources in such networks (such as bandwidth and battery life) are constrained. These issues make routing packets between any node pair a challenging task, especially for proactive MANET routing protocols. Each node of a proactive MANET routing protocol maintains routing information to every other node in the network at all times. The routing information has to be updated to reflect the topology changes and guarantee the correctness of route selection. Additionally, the dissemination of control messages has to be optimised for efficient resource usage and to alleviate channel contention problems.

This dissertation investigates the signalling performance of proactive MANET routing protocols, using a combination of simulation-based study and model-based analysis. The impacts of soft state signalling, especially the refresh intervals, are studied under various scenarios. A variety of topology advertisement strategies are presented. Two optimised neighbour detection schemes are proposed, namely the Dynamic Timer algorithm and the Fast Neighbour Handshake algorithm, in order to enhance routing performance.

These efforts have allowed this dissertation to provide a clear insight into various aspects of soft-state signalling performance in dynamic resource-constrained networks, and to provide useful understanding on how to effectively design MANET protocols. In addition, this research proposes a number of original signalling mechanisms to improve routing performance. These are described and evaluated within the dissertation along with recommendations for further study.
Acknowledgements

I would like to thank my advisors Søren-Aksel Sørensen and Saleem Bhatti for providing the advice, support and encouragement throughout the years. Soren has been inspiring me to carry out rigorous high-quality research. Discussions with Soren have frequently led to key insights both in my work and in this dissertation. Saleem has been giving steady influences throughout my PhD study. He has always been patient and encouraging in times of new ideas and difficulties. I have benefited tremendously from observing him and conducting research with him, and the flexibility he permitted in choosing and pursuing research ideas. After all, I have seen in them excellent advisors who can bring the best out from their students, and outstanding researchers who are capable of choosing and approaching compelling research problems.

I would like to acknowledge the financial support from UCL and University UK during the last four years. This work is impossible, without the Graduate School Research Scholarships (GSRS) from UCL Graduate School and the Overseas Research Students (ORS) Awards from University UK. In addition, I would also like to thank the Computer Science Department for its generous financial support for attending conferences.

It has been a privilege and an invaluable experience for me to participate in the Madeira project at Ericsson Research Ireland. This is especially so because it is not everyday that a PhD student gets to learn from and interact with researchers in a dynamic and stimulating industrial environment. I have greatly enjoyed the opportunity to work with Alan Leddy, Sandra Collin, Liam Fallon and David Cleary. Daryl Paker has been a pleasure to work with. He has been a faithful friend and co-author over the past several years. Long discussions with Francoise Sailhan and Samir Ghamri-Doudane have significantly improved my work and inspired many new research directions.

I would like to thank Angela Sasse for her support in providing fundings for a range of language courses when I started my PhD, and her efforts in organising PhD seminars. I also want to thank Damon Wischik, whose mathematical understanding is tremendous and whose scientific work inspired me a lot. My work has greatly benefited from suggestions and kind encouragement from Mark Handley and Stephen Hailes during my first-year and transfer vivas. I am particularly thankful to Adam Greenhalgh for his hard work on maintaining the experimental testbed. I have been very lucky to have discussions with Manish Lad. Thank him for his support and encouragement.

I am also indebted to the other members of Network Research Group. I want to thank all the members (present and past) for their support and for providing a pleasant and productive working atmosphere:
Peter Kirstein, Jon Crowcroft, Sheng Jiang, Soo-Hyun Choi, Michael Rogers, Piers O’Hanlon, Dimitrios Miras and Theo Pagtzis. Very special thanks go to Richard Smith. It was the most pleasant time to work with Richard on the GRS project.

Finally, I would like to thank my wife Ruping Liu for her love and encouragement. And, thank her for her support when I have needed it the most. I would also like to thank my mother, father, and sister for their infinite support throughout everything.
Contents

1 Introduction .......................................................... 16
  1.1 Problem Statement ........................................... 17
  1.2 Dissertation Statement ..................................... 19
  1.3 Contributions ................................................ 19

2 Effective Routing State Maintenance in MANETs ........ 21
  2.1 MANET Overview ............................................ 21
  2.2 MANET Concepts ............................................ 23
  2.3 Signalling in Proactive MANET Routing Protocols .... 24
  2.4 Motivation .................................................... 24
    2.4.1 General Concerns ...................................... 25
    2.4.2 Concerns of MANET Signalling .................... 26
  2.5 Fundamental Challenges ................................... 27

3 Background and Related Work ................................. 31
  3.1 Network Signalling .......................................... 31
    3.1.1 Signalling Concepts .................................... 31
    3.1.2 Signalling in Wireless Networks ................. 32
    3.1.3 Soft State Signalling .................................. 33
  3.2 Routing Protocols for Mobile Ad Hoc Networks ...... 35
    3.2.1 Traditional Proactive Routing: LS vs. DV ...... 35
    3.2.2 Proactive Routing ....................................... 36
    3.2.3 Reactive Protocols ...................................... 41
    3.2.4 Characterisation of MANET Routing Dynamics .. 44
    3.2.5 Performance Comparison of Routing Protocols ... 44
  3.3 Signalling in Proactive Routing Protocols ............... 45
    3.3.1 HELLO Based Neighbour Detection ............... 45
    3.3.2 Cross-layer Neighbour Detection ................. 45
    3.3.3 Topology Update Strategies ....................... 46
    3.3.4 Adaptive Signalling ................................... 46
  3.4 Credibility of MANET Simulation-based Studies ....... 47
4 Experimental Methodology

4.1 Methodology of Performance Evaluation

4.2 Mobility Models

4.2.1 Random Waypoint Model & Its Variants

4.2.2 Random Trip Mobility Model

4.2.3 Manhattan Model

4.2.4 Reference Point Group Model

4.2.5 Mobility Models Based on Social Network Theory

4.3 Simulation Environments

4.3.1 Choice of Simulation Tools

4.3.2 Choice of Application Scenarios

4.3.3 Choice of Mobility Models

4.3.4 Design of Traffic Patterns

4.3.5 Performance Metrics

4.3.6 Assumptions

4.4 Data Collection, Processing and Analysis

4.4.1 Data Collection

4.4.2 Data Processing

4.4.3 Data Analysis

4.5 Factorial Analysis

4.5.1 Factors

4.5.2 Factor Selection

4.6 Summary

5 Understanding the Impact of Soft Intervals on the Performance of OLSR and DSDV

5.1 Overview

5.2 Soft State Signalling in Proactive MANET Routing Protocols

5.2.1 Optimised Link State Routing

5.2.2 Destination-sequenced Distance Vector Routing

5.3 A Soft-State Model Based on Probability Theory

5.3.1 Soft State Model without Channel Loss

5.3.2 Soft State Model with Channel Loss

5.3.3 Analysis

5.4 Simulation

5.4.1 Set-up

5.4.2 Scenarios

5.4.3 Metrics
Contents

5.5 Observations ......................................................................................................................... 76
  5.5.1 Observations under Synthetic Scenarios - OLSR .................................................. 76
  5.5.2 Observations under Concrete Scenarios - OLSR ............................................. 79
  5.5.3 Observations - DSDV .............................................................................................. 80
5.6 Summary ................................................................................................................................ 81

6 Topology Control and Maintenance for Mobile Ad Hoc Networks Using OLSR 82
  6.1 Temporal Redundancy .......................................................................................................... 84
    6.1.1 Overview .................................................................................................................... 84
    6.1.2 Topology Maintenance Strategies in Proactive MANET Routing Protocols ......... 85
    6.1.3 Model Based Analytic Study .................................................................................... 86
    6.1.4 Simulation Based Performance Evaluation ........................................................... 89
    6.1.5 Observations .............................................................................................................. 90
  6.2 Spatial Redundancy .............................................................................................................. 92
    6.2.1 Overview .................................................................................................................... 92
    6.2.2 Protocol Overview .................................................................................................... 93
    6.2.3 Proposed Spatial Redundancy Options ..................................................................... 94
    6.2.4 Simulation Based Performance Evaluation ............................................................ 95
    6.2.5 Observations .............................................................................................................. 96
  6.3 Summary ...................................................................................................................................100

7 Effective Neighbour Sensing Mechanisms for OLSR and DSDV 102
  7.1 Dynamic Timer Algorithm ......................................................................................................103
    7.1.1 Problem Formalisation ..............................................................................................104
    7.1.2 Proposed Dynamic Timer Algorithm ........................................................................105
  7.2 Fast Neighbour Handshake Scheme .......................................................................................110
    7.2.1 Explicit Handshake Scheme .......................................................................................111
    7.2.2 Analysis .....................................................................................................................112
  7.3 Performance Evaluation .............................................................................................................116
  7.4 Observations .............................................................................................................................117
    7.4.1 Performance of Dynamic Timer Algorithms ...............................................................117
    7.4.2 Performance of Fast Handshake Algorithms ..............................................................121
  7.5 Summary ...................................................................................................................................122

8 Conclusions and Future Work 124
  8.1 Discussion .................................................................................................................................124
  8.2 Conclusions ...............................................................................................................................127
  8.3 Future Directions .......................................................................................................................128

Appendices 129
List of Figures

2.1 A Multi-hop Ad-hoc Network .............................................................. 22

4.1 Object Binding between OTcl and C++ in NS .............................................. 53

4.2 Impacts of Signalling on End-to-End Delay ................................................. 58

4.3 Standard Trace Example .............................................................................. 60

4.4 Customised Trace Example ......................................................................... 60

5.1 Soft State Update without Channel Loss .................................................... 71

5.2 Inconsistency Ratio vs. Refresh Interval (without Channel Loss) ....................... 72

5.3 Soft State Update with Channel Loss ......................................................... 72

5.4 Inconsistency Ratio vs. Refresh Interval (with Channel Loss) .......................... 74

5.5 Impact of HELLO Intervals on OLSR Throughput \((t=5, \tau_r=250)\) .................. 77

5.6 Impact of TC Intervals on OLSR Throughput \((h=2, \tau_r=250)\) ..................... 77

5.7 Impact of Refresh Intervals on OLSR Throughput \((n=20, \tau_r=125)\) .............. 77

5.8 Impact of HELLO Intervals on OLSR Overhead \((t=5, \tau_r=250)\) .................... 78

5.9 Impact of TC Intervals on OLSR Overhead \((h=2, \tau_r=250)\) ....................... 78

5.10 OLSR Throughput vs. Node Velocity \((t=5, \tau_r=250)\) .............................. 79

5.11 OLSR Throughput vs. Node Velocity \((h=2, \tau_r=250)\) .............................. 79

5.12 Impact of Update Intervals on DSDV Performance \((n=20, \tau_r=250)\) ............. 80

5.13 Impact of Update Intervals on DSDV Performance \((n=50, \tau_r=250)\) ............. 81

6.1 Impact of Topology Update Interval on Consistency .................................... 87

6.2 Control Overhead under Proactive and Reactive Options ............................... 88

6.3 Low-Density Network vs. High-Density Network \((\tau_r=250)\) ......................... 89

6.4 OLSR Throughput vs. Topology Update Intervals \((h=2, \tau_r=250)\) ............... 91

6.5 OLSR Control Overhead vs. Topology Update Intervals \((h=2, \tau_r=250)\) ........... 91

6.6 OLSR Routing Performance under Reactive Topology Update Options \((h=2, \tau_r=250)\) .......................................................... 92

6.7 Amount of OLSR Control Overhead under \(pta.d\), \(pta.u\) and \(ft.s\) \((h=1, t=5, \tau_r=250)\) .......................................................... 97

6.8 Number of OLSR Control Packets under \(pta.d\), \(pta.u\) and \(ft.s\) \((h=1, t=5, \tau_r=250)\) .......................................................... 97

6.9 Size of OLSR Control Packets under \(pta.u\) and \(ft.s\) \((h=1, t=5, \tau_r=250)\) .................. 98

6.10 OLSR Throughput under \(pta.d\) and \(pta.u\) with Cross Layer Support \((h=1, t=5, \tau_r=250)\) .......................................................... 98

6.11 OLSR Throughput under \(pta.d\) and \(ft.s\) with Cross Layer Support \((h=1, t=5, \tau_r=250)\) .......................................................... 99
List of Figures

6.12 OLSR Throughput under $p_{ta.d}$ and $p_{ta.u}$ without Cross Layer Support ($h=1 \ t=5 \ r=250$) 99
6.13 OLSR Throughput under $p_{ta.d}$ and $f_{ts}$ without Cross Layer Support ($h=1 \ t=5 \ r=250$) 100

7.1 Link Change Rate vs. Node Velocity ........................................ 107
7.2 Expected Refresh Interval vs. Link Change Rate ............... 110
7.3 HELLO Based Neighbour Detection Process ...................... 113
7.4 Fast Neighbour Handshake Process ........................................ 114
7.5 $\Delta_{t_p}^i$ vs. $\lambda$ ($\tau=2 \ \Delta_t =1.5$) .................. 115
7.6 $\Delta_{t_p}^i$ vs. $r$ ($\lambda=2 \ \Delta_t =1.5$) .................. 115
7.7 OLSR Performance under DT_MIAD ($n=20 \ \alpha=2 \ \beta=0.02$) .......... 117
7.8 OLSR Performance under DT_MIAD ($n=50 \ \alpha=2 \ \beta=0.02$) .......... 118
7.9 DSDV Performance under DT_MIAD ($n=20 \ \alpha=2 \ \beta=0.1$) .......... 118
7.10 DSDV Performance under DT_MIAD ($n=50 \ \alpha=2 \ \beta=0.1$) .......... 118
7.11 OLSR Performance under DT_ODPU ($n=20$) .................... 119
7.12 OLSR Performance under DT_ODPU ($n=50$) .................... 120
7.13 DSDV Performance under DT_ODPU ($n=20$) .................... 120
7.14 DSDV Performance under DT_ODPU ($n=50$) .................... 120
7.15 OLSR Throughput with Fast Handshake Algorithms ($h=2 \ t=5 \ r=250$) .......... 121
7.16 OLSR Overhead with Fast Handshake Algorithms ($h=2 \ t=5 \ r=250$) .......... 122

8.1 Relationship between Chapter 5, 6 & 7 ......................... 125
8.2 Impact of Refresh Interval vs. Change Rate ............... 126

B.1 PDF of Route Duration for Random Trip Model ($n = 20$) .......... 134
B.2 PDF of Route Duration for Random Trip Model ($n = 50$) .......... 134
B.3 CDF of Route Duration for Random Trip Model ($n = 20$) .......... 134
B.4 CDF of Route Duration for Random Trip Model ($n = 50$) .......... 135
B.5 PDF of RD with Specific Route Lengths ($n = 20$) .......... 137
B.6 PDF of RCI for Random Trip Model ($n = 50$) .......... 137
B.7 Average Link/Route Duration (Random Trip Model) .......... 138
List of Tables

4.1 MAC/PHY Modules .......................................................... 52
5.1 Notation in Soft State Model ........................................... 71
5.2 MAC/PHY Layer Configurations ........................................ 75
6.1 Model Variables ............................................................ 86
6.2 MAC/PHY Layer Configurations ........................................ 90
7.1 Notation in Problem Formalisation .................................... 104
7.2 Notation in Link Change Rate Equation .............................. 106
7.3 DT-MIAD Notation .......................................................... 107
7.4 Notation in Neighbour Detection Analysis ......................... 113
7.5 MAC/PHY Layer Configurations ........................................ 116
7.6 OLSR Parameters .......................................................... 116
7.7 DSDV Parameters .......................................................... 116
A.1 Wireless Trace Format ..................................................... 130
B.1 Symbols .......................................................................... 132
B.2 Goodness of Fit (Fig B.1(a)) ............................................. 135
B.3 Goodness of Fit (Fig B.1(b)) ............................................. 136
B.4 Goodness of Fit (Fig B.2(a)) ............................................. 136
B.5 Goodness of Fit (Fig B.2(b)) ............................................. 136
B.6 Goodness of Fit (Fig B.5(a)) ............................................. 136
B.7 Goodness of Fit (Fig B.5(b)) ............................................. 136
B.8 Goodness of Fit (Fig B.6(a)) ............................................. 138
B.9 Goodness of Fit (Fig B.6(b)) ............................................. 138
Availability

The detailed implementations of the proposed algorithms in this dissertation are available on-line.

The implementation of soft-state signalling is available from
http://www.cs.ucl.ac.uk/staff/y.huang/softstateMANET

The implementation of topology temporal-redundancy advertisement options is available from
http://www.cs.ucl.ac.uk/staff/y.huang/topo_temporal

The implementation of topology spatial-redundancy advertisement options is available from
http://www.cs.ucl.ac.uk/staff/y.huang/topo_strategy

The implementation of fast handshake algorithms for neighbour detection, including BHS (Broadcast based Handshake) algorithm and UHS (Unicast based Handshake) algorithm, is available from
http://www.cs.ucl.ac.uk/staff/y.huang/fh

The implementation of dynamic timer algorithms for neighbour detection, including DT_MIAD (Dynamic Timer Based on Multiplicative Increase Additive Decrease) and DT_ODPU (Dynamic Timer Based on On-Demand Proactive Update), is available from
http://www.cs.ucl.ac.uk/staff/y.huang/dt

The scripts used in data analysis, including python scripts and shell scripts, are available from
http://www.cs.ucl.ac.uk/staff/y.huang/scripts

The original data collected in the simulations also appears with the implementations above.


Chapter 1

Introduction

State, as interpreted in dictionaries, refers to mode or condition of being. In the context of IP networks, state is to record conditions of applications or operations, in order to provide continuous functionalities and services. State can be per-flow based, such as resource reservation state of the Resource Reservation Setup Protocol (RSVP) [1, 2, 3] and session information of the Session Initiation Protocol (SIP) [4, 5, 6]. It can also be network control state, such as route state of the routing protocols, which is not associated with any specific data flows. In order to ensure adequate service provision (i.e. establishing, maintaining and tearing down network services), control messages are exchanged between network entities to derive and maintain state information, keeping state updated when the topology or activity of the network changes. This is called (message-based) signalling.

Based on its locations, state associated with network applications and services is categorised into soft state and hard state [7].

The state maintained in the network is soft state, in the sense that it can be quickly reconstructed based on periodic updates as the network topology changes (due to routers and switches going on and off line). The state is expected to be self-healing in presence of network failures, so that the loss of the state would not result in more than a temporary loss of service given that connectivity exists. Soft-state signalling sends out best-effort, periodic refresh messages to maintain state information. State would eventually expire and be removed unless periodically refreshed by the receipt of a signalling message indicating its validity. Accordingly, soft-state signalling requires no explicit state removal. Since message loss or state corruption could be restored by subsequent refresh messages, reliable message delivery is not required. Thus, soft-state signalling is simple and robust. However, the volume of the state information and manual configuration related to state manipulation needs to be minimised in order to reduce the control overhead.

Hard state takes the inverse approach to soft state. In hard-state signalling, state remains installed unless explicitly removed by the receipt of a state-teardown message. So, hard-state signalling requires a mechanism to remove orphaned state that remains after the state-installer has crashed or departed without removing state. Similarly, since state installation and removal are performed only once, reliable message delivery is thus necessary to maintain consistency.

In recent years, there has been a growing interest in signalling protocols and their impacts on system
1.1. Problem Statement

performance. For example, RSVP was originally designed as a signalling protocol for the Integrated Services (IntServ) QoS model [8]. However, RSVP is not limited to carrying QoS-related messages. Its generic aspect has enabled it to carry different information to meet different application requirements, such as Multi-Protocol Label Switching (MPLS) Label-Switched Path (LSP) signalling [9]. In addition, the popularity of RSVP has spurred further investigations on its performance [10, 11, 12]. Komolafe and Sventek evaluated the performance of RSVP control message delivery mechanisms in maintaining reservation state in [10] and analysed a number of parameters that affected the efficiency of RSVP in establishing and maintaining reservations in [11].

Most of these studies on signalling are based on wired networks. With the fast development and growth of the Internet, especially the wide deployment of high-capacity optical media and routing/switching devices, the basic reliability and robustness of wired links has improved considerably. The end nodes are therefore responsible for maintaining the integrity of the communication, as the end-to-end principle implies [7, 13].

In contrast, mobile wireless networks consist of largely autonomous mobile nodes with multiple wireless link options. The reliability of wireless links is governed more strongly by the basic physics of the medium, the instantaneous radio propagation conditions and node movement. Node mobility may lead to the breakage of the established routes and network topology may change randomly and rapidly. Moreover, nodes are organised in a decentralised infrastructure, which increases significantly the difficulties in designing efficient signalling protocols.

Such networks are highly resource-constrained. Links of a wireless network typically have lower capacities than those of fixed networks, and the available link capacity of a wireless channel is often much less than the channel capacity. Other resources, such as battery life, are usually limited for a variety of MANET applications, such as sensors and personal devices.

In short, state maintenance in such networks is essentially more challenging than in wired networks. The following section illustrates the problems of maintaining routing state in mobile ad hoc networks (MANETs).

1.1 Problem Statement

Proactive MANET routing protocols are usually based on either the link state (LS) (i.e., Dijkstra) or the distance vector (DV) (i.e., Bellman-Ford) algorithms [14, 15]. In these protocols, each node maintains routing information to every other node in the network at all times. The routing information can be either topological repositories (LS protocols) or distance to other nodes (DV protocols). The routing protocols ensure that all nodes at all times have sufficient and correct topological information in order to compute routes to all destinations in the network. Due to the frequent topology changes caused by mobility, the routing information has to be updated to reflect the topology changes and guarantee the correctness of route selection. This requires the nodes of the proactive routing protocol detect link changes quickly and broadcast topology updates with little delay. Such neighbour detection and topology advertisement mechanisms are mainly through exchanging messages between nodes, i.e. signalling.

The features of soft-state signalling, simplicity and robustness, make soft-state an attractive choice
for state maintenance in MANETs. In MANET routing protocols, periodic HELLO messages are sent locally to detect link changes, including link establishment, and to maintain node status. In addition, proactive routing protocols like the Optimised Link State Routing (OLSR) protocol [16] propagate periodic network-wide topology update messages to advertise topology changes.

Despite its fundamental importance, there is still no comprehensive study on the performance of signalling mechanisms in MANETs. Many properties of soft-state signalling are not yet fully understood, especially their impacts on routing performance and the circumstances in which it might best be employed in the context of MANETs. In particular, the following questions have not been well addressed in previous studies:

- Could periodic topology updates alone be effective enough to capture topology changes?
- How could the value of the refresh intervals be determined in order to achieve the best balance between routing performance and control overhead, especially in presence of tight resource constraints?
- How much could the routing performance be improved if the refresh interval is reduced by 1s?
- Could the routing performance be improved if extra topology state is advertised and maintained by each node?
- How could the scalability of signalling protocols be achieved, especially with the increase in number of nodes and node mobility?
- How could topology advertisement strategies be optimised in order to reduce unnecessary resource consumption, without adversely affecting the routing performance?
- How could signalling protocols adjust their behaviours when propagating messages across different wireless networks?

Some of these problems are generally concerned with existing signalling approaches designed for IP networks, while others are specific to MANETs. However, all are applicable to MANETs.

To summarise, despite numerous performance studies on MANET routing protocols, it is still crucial to analyse the performance of signalling mechanisms in the presence of frequent topology changes, and re-evaluate fundamental protocol design issues. Such analysis and re-evaluation are critical to understand how the signalling mechanisms behave in proactive MANET routing protocols and how to design effective signalling protocols in MANETs.

This dissertation concentrated on the problem of effective routing state maintenance in MANETs running proactive routing protocols. It investigated the issue of routing data packets reliably through intermittently connected networks without a priori knowledge of node movements or locations. The proposed signalling solutions aim at reducing data packet losses by reducing failure detection latency and adaptation latency to topology variations, and consequently providing improvements in achieving a reliable, scalable and robust data delivery service. OLSR (Optimised Link State Routing) and DSDV
1.2. Dissertation Statement

A performance analysis of soft-state signalling protocols in MANETs reveals the impact of soft state refresh intervals depends on a variety of factors. Effective signalling can improve packet delivery performance of proactive MANET routing protocols without introducing significant control overhead.

1.3. Contributions

In this dissertation, signalling performance has been examined, especially for soft-state protocols, across mobile ad hoc networks. Based on the performance evaluation studies, a variety of novel signalling solutions have been proposed in order to improve routing performance. The specific contributions are:

- **A performance evaluation of Soft State signalling in MANETs.** The impact of soft state signalling on routing performance is investigated under various conditions (through varying node density and node velocity). A novel soft-state model based on probability theory is presented to give an in-depth analysis of the observations. More details can be found in Chapter 5.

- **An efficient mechanism for topology advertisements.** The novelty of this approach is to analyse the efficiency of topology advertisements from the following two aspects: temporal redundancy (i.e. adjusting the topology update frequency) and spatial redundancy (i.e. using extra topology state information in each topology control message). A quantitative analysis of the impact of topology advertisements on OLSR routing performance is presented, which gives a better understanding of the topology update strategies in MANETs using OLSR. This work can be found in Chapter 6.

- **The creation of novel neighbour detection schemes.** Two neighbour detection schemes are proposed to improve throughput and lower control overhead, namely the Dynamic Timer algorithm (an adaptive proactive routing algorithm that varies the frequency of neighbour detection messages in response to network load and mobility conditions), and the Fast Neighbour Handshake algorithm (a fast neighbour sensing scheme that uses explicit handshake mechanism in neighbour detection).

  The Dynamic Timer algorithm helps achieve a balance between routing performance and overhead cost, while the Fast Neighbour Handshake algorithm reduces neighbour discovery latency and improves route availability. More detail concerning these algorithms is given in Chapter 7.

In addition to solving the specific problems in MANETs discussed above, this dissertation also derives a set of general principles and lessons to facilitate the design of protocols and applications for mobile wireless networks. These include:

- **Trade-offs versus optimisation.** This dissertation re-evaluates the trade-off between routing performance and control overhead from the aspect of signalling. Through performance studies on
the impact of refresh intervals (see Chapter 5), this dissertation reveals its dependence on various factors. Further optimisation methods and their benefits in improving routing performance (see Chapter 6 & 7) support the results of the performance studies.

- **Adaptability without dependencies.** Adapting the refresh intervals of soft-state signalling to node movement could help balance routing performance and control overhead. However, the performance of existing adaptive routing approaches [17, 18] largely depends on the accuracy of network measurement. Such a dependency jeopardises the applicability of these algorithms. The adaptive algorithms proposed in Chapter 7 eliminate such dependencies by introducing and tracking internally maintained variables. In the solutions proposed in this dissertation, the adaptation to node movement is fully automated.

- **Cross-layer protocol optimisations.** This dissertation refines the idea of cross-layer protocol optimisations as a design methodology (by which information is exposed across multiple layers of the conventional protocol stack) to optimise MANET routing performance [19, 20], and to consider the interdependence between cross layer optimisations and factors such as traffic load. For example, through using the failure events of data packet dissemination, link layer notification improves neighbour detection. However, spatial topology redundancy (see Chapter 6) provides extra routes for data traffic and reduces average data rate per link, which weakens the improvements brought by cross-layer notification. Another example is with link-layer notification; the unicast handshake option helps detect link breakage (see Chapter 7). However, due to the queue length limitations, the ARP (Address Resolution Protocol) overhead introduced by the UHS option leads to data packet drops, which lowers the overall performance.

In summary, this dissertation has analysed a number of factors contributing to MANET signalling performance and identified concepts that promote understanding of various aspects of signalling performance in dynamic and resource-constrained networks. The focus has been on verifying some fundamental observations and putting forward findings that are missing or only partially addressed in existing MANET performance research efforts. Such a reconsideration of the basic mechanisms and techniques is critical to today's wireless network research.
Chapter 2

Effective Routing State Maintenance in MANETs

This chapter first presents an overview of MANETs, followed by a brief introduction of the roles of routing state maintenance in MANETs. Further to Chapter 1, this chapter illustrates the major concerns of existing signalling mechanisms, by which this study has been motivated. Finally this chapter identifies the fundamental challenges of routing state maintenance, with in-depth analysis of the contributions to make it more effective.

2.1 MANET Overview

MANETs [21] are self-organising multi-hop wireless networks (see Fig 2.1), consisting of mobile nodes connected by wireless links. The network may operate in an isolated manner, or be connected to a fixed network (such as the Internet) through gateways. The logical nodes of such networks can be various objects, from very large objects (such as air planes, ships, trucks and cars), to very small objects (such as sensors). Correspondingly, the capacity and resource availability of the nodes vary significantly.

The nodes are usually equipped with wireless transmitters and receivers using antennas that may be omnidirectional, highly-directional or steerable. Therefore, the physical wireless connectivity between any nodes depends on various factors such as node positions, radio ranges, interference levels and external environmental circumstances that might affect signal transmissions. Node mobility imposes further communication challenges since the established routes may be unstable and even get broken.

The main characteristics of MANETs include:

- **Dynamic Topologies.** The nodes in MANETs are free to move arbitrarily. Also, the nodes might get powered-off when running out of battery or if re-started. Thus, network topology may change randomly and rapidly (see Appendix B). For this reason, traffic metrics, closely correlated with network connectivity, are usually dynamic and subject to topology changes. MANET protocols and applications have to be resilient and robust in the presence of frequent topology changes.

- **Resource Constraints.** Links of a wireless network typically have lower capacities than those of fixed networks. The available link capacity of a wireless channel is often much less than the channel capacity because of the effects of fading, noise, interference and MAC operation. This makes
network congestions more common than fixed networks. Other resources, such as battery life, are usually limited for a variety of MANET applications, such as sensors and personal devices. Therefore, in order to provide similar services to the fixed network infrastructure, there are increasing demands for MANET protocols to be well designed to optimise resource usage. Such demands may intensify with the deployment of real-time applications, e.g. multimedia application, in ad hoc environments such as sensor networks.

It is worth mentioning that, resource constraints of MANETs, although inherited from wireless networks, are quite different in handling. Common wireless networks, either with stationary nodes or with certain infrastructure, or both, can optimise control overhead by adopting hierarchical design, which could partially solve the problem of resource constraints. In a mobile ad hoc network, because of node mobility and their ad hoc nature, flooding is the generally applicable method in topology advertisements. Consequently, resource constraints present more challenges in designing MANET signalling protocols.

- **Decentralised Infrastructure.** Due to the dynamic nature, MANET nodes are organised in decentralised infrastructure. In each network, there is no central control node responsible for packet forwarding and node discovery. Such a design increases the resilience in presence of failure. On the other hand, this increases the difficulties in designing effective and efficient protocols and applications. For example, because of the decentralisation of MANETs, the nodes have to broadcast or flood requests in route or service discovery, which may introduce excessive control overhead and increase channel contention, if not properly configured.

The characteristics listed above bring several performance concerns for existing design principles of MANET signalling protocols, most of which are extended from general design principles of the Internet. Before discussing the major concerns and the challenges of soft-state signalling, this dissertation presents some details of the signalling mechanisms in MANET routing protocols.
2.2 MANET Concepts

This section briefly describes several basic concepts of proactive MANET routing protocols. More details about these protocols can be found in Chapter 3.

**MANET Nodes.** Each node in a MANET runs one or multiple routing protocols and is prepared to forward data packets.

**Wireless Channel.** A subdivision of the physical wireless medium that allows possibly shared independent uses of the medium. Channels may be made available by subdividing the medium into distinct time slots, distinct spectral bands, or decorrelated coding sequences [22].

**Flooding.** The process of delivering data or control messages to every node within the ad hoc network [22].

**Neighbour Node.** A node \( j \) is a neighbour node of node \( i \), if node \( i \) can hear node \( j \) on some interfaces. Neighbour node is not a symmetric property.

**Routing State.** In proactive MANET routing protocols, each node maintains routing information to every other node in the network at all times. The routing information can be either topological repositories (LS protocols) or distance to other nodes (DV protocols).

**Routing State Maintenance.** Due to node mobility, the routing state has to be updated to reflect the topology changes. This requires the nodes of proactive routing protocols detect neighbour changes quickly and broadcast topology updates with little delay.

**Neighbour State.** Each node in a MANET keeps track of the status of its neighbours, including arrival of new neighbours and loss of existing neighbours. Neighbour state can be either stored in a separate neighbour repository (LS protocols), or maintained in routing tables as one-hop destination nodes (DV protocols). Each neighbour entry contains the valid time during which the neighbour is considered as available and useable.

**Neighbour Detection.** Due to node mobility, neighbour state needs to be updated frequently to reflect neighbour changes. The process of updating neighbour state is called neighbour detection. It includes discovering new neighbours and detecting neighbour loss.

**Link State.** Link state refers to the connectivity status between the nodes and their neighbours. It can be either uni-directional or bi-directional. Each link state entry contains the interface addresses of the local node and the neighbour node (i.e. the end points of a link), and the valid time during which the link is considered as available and useable.

**Link Sensing.** Link state needs to be updated frequently to reflect the changes in link status. Link sensing is to update link state, removing expired link state entries and creating new entries. There are mainly two link-sensing methods in existing MANET routing protocols: (1) exchanging periodic HELLO messages, (2) cross-layer notification.

**Topology Repository.** In LS protocols, topology repository maintains the topology information of the whole network. This information is acquired from topology update messages and is used for routing table calculations.

**Topology Update.** Topology update (or advertisement) is the process of advertising the detected
neighbour (or link) changes to every other nodes in the network. In LS protocols, each node sends topology advertisements to every neighbour. In turn, each received topology advertisement is copied and forwarded to every neighbour except the one that sends it. In DV protocols, the nodes exchange routing tables with their neighbours, instead of flooding topology information through the network.

2.3 Signalling in Proactive MANET Routing Protocols

Two features of soft-state signalling, *simplicity* and *robustness*, make soft-state an attractive choice for a variety of network protocols and applications, especially for dynamic networks such as MANETs.

- **Neighbour Detection.** In most MANET routing protocols, either proactive protocols like OLSR (Optimised Link State Routing protocol) [16], DSDV (Destination-Sequenced Distance-Vector Routing) [23], TBRPF (Topology Broadcast based on Reverse-Path Forwarding) [24] or reactive protocols like AODV (Ad hoc On-Demand Distance Vector Routing) [25], TORA (Temporally-Ordered Routing Algorithm) [26], or even hybrid protocols like ZRP (Zone Routing Protocol) [27], periodic HELLO messages are exchanged between neighbouring nodes to detect link dynamics and to maintain node status. For example, nodes running OLSR [16] discover new neighbours and links when receiving the first HELLO message from an unknown neighbour. Moreover, obsolete neighbour states (caused by for example link breakage) are removed after state time-out.

- **Topology Advertisements.** Proactive routing protocols like OLSR [16], TORA [26] and TBRPF [24] propagate periodic network-wide topology update messages to advertise topology changes. In addition to initiating new link state in topology repositories of each node, for example, the topology advertisement process in OLSR removes obsolete topology state either implicitly by assigning sequence numbers to topology advertisements, or by state time-out.

  Soft-state signalling plays an important role for MANET routing performance.

  Due to node mobility or failure, the established links maybe get broken. If nodes are unaware of such changes and keep forwarding data packets towards these broken links, the data packets could be lost. In addition, if the nodes with broken links have detected the link breakage but fail to notify other nodes in the network, the routing state repositories across the nodes could be inconsistent. One possible result of route inconsistency is routing loops.

  Reducing refresh intervals in neighbour detection leads to smaller link detection latency, while reducing refresh intervals in topology advertisements results in shorter delay in link breakage notification. As a result, there should be fewer data packet loss, which improves routing throughput.

2.4 Motivation

Despite the fundamental importance of signalling, there is still no study on the performance of soft state protocols in MANETs. Many properties of soft state signalling are not yet fully understood in the context of MANETs, especially their impacts on routing performance and the circumstances in which it might best be employed. This section gives a detailed discussion on the major concerns related to soft-state signalling approaches.
2.4. Motivation

2.4.1 General Concerns

Carpenter [28] described several basic requirements of state maintenance in Internet infrastructure:

"... This state must be self-healing; adaptive procedures or protocols must exist to derive and maintain that state, and change it when the topology or activity of the network changes. The volume of this state must be minimised, and the loss of the state must not result in more than a temporary denial of service given that connectivity exists. Manually configured state must be kept to an absolute minimum."

In particular, the general concerns of the soft state approach are listed as follows.

- **Configuring Timer Intervals.** Timer interval in soft-state signalling has to be set manually by administrator. The value is mainly determined based on recommendations of original protocol designers or empirically. Usually there is no careful calculation, solid theoretical studies or experimental research on how to configure the intervals of various soft-state timers, including refresh timers and time-out timers. Consequently, as an arbitrary choice, different timers often use the same (fixed) interval. Moreover, the value of such intervals usually remains fixed no matter what network conditions (i.e. node velocity, link loss rate) are. Therefore, several questions as to the configuration of timer intervals may arise: e.g. does the default value of timer intervals work well against all types of link failures under various scenarios? Given requirements on system consistency, how do we determine the value of the refresh intervals in order to achieve the best balance between performance and overhead?

- **Trade-off between Performance and Overhead.** It is commonly believed that a smaller refresh interval in a soft state mechanism could speed up adaptation to changes at the expense of increased overhead. However, there is no solid study on *how much* it could improve the consistency and the amount of overhead. Especially, this question is critical to signalling in MANETs. On one hand, topology changes require effective signalling to maintain routes and so maximise the throughput. On the other hand, the resource constraints of MANETs require minimum control overhead to reduce channel contention and battery consumption. With existing signalling approaches, it is a challenge to balance signalling performance and overhead.

- **Network Heterogeneity.** The convergence of different classes of wireless networks, including in-room Infrared networks, building-wide WLAN and cellular networks, blurs their distinctions. The devices in the networks have diverse capacities such as bandwidth and battery life. Therefore, signalling protocols need to adjust their behaviours when propagating messages across these devices. In addition, the state description in signalling should be generic to provide additional flexibility in defining the objects carried by signalling messages. Existing MANET signalling approaches have not provided any solutions to these issues.

These problems are generally concerned with existing signalling protocols and mechanisms designed for IP networks with relatively stable and fixed connections [29]. Furthermore, the characteristics of MANETs introduce some new requirements for signalling protocols.
2.4.2 Concerns of MANET Signalling

In the context of MANETs, there have been the following concerns about soft-state performance, especially for proactive routing protocols.

- **Intensified Tussles between Performance and Overhead.** Due to topology changes caused by mobility, the route states have to be updated frequently in order to reflect the topology changes and strive for the correctness of route selection. Because of the decentralised infrastructure, MANETs have to rely on flooding based route (or service) discovery/maintenance mechanisms, which generates control overhead in the order of $o(n^2)$ ($n$ node population).

- **Scalability.** One contributing factor of scalability for soft-state protocols is the potentially excessive routing message overhead caused by the increase of network population and node mobility. For example, in a network with population $n$, topology updates of LS (Link State) protocols generate routing overhead in the order of $o(n^2)$. In large networks with high mobility, the transmission of routing information will ultimately consume a large proportion of the bandwidth and consequently congest the link channel, rendering it unfeasible for bandwidth limited wireless ad hoc networks.

The amount of routing state, i.e. the size of routing tables and topology repositories, is another concern for MANET signalling. In proactive routing protocols, large routing tables or topology repositories imply heavier processing overhead in route calculation, or larger control packet size, hence increased control overhead.

Compared with common wireless networks, the scalability problem of MANET signalling is more challenging because of the presence of both large node population and node mobility. If nodes are stationary, the scalability problem can be effectively handled with a conventional hierarchical routing approach. In contrast, with node mobility, the hierarchical partitioning must be continuously updated, which makes the hierarchical design inefficient in MANET scenarios.

- **Effectiveness of Topology Advertisements.** Considering the amount of control overhead, topology advertisement intervals are usually set to relatively large value, for example, 5s in OLSR. In a high-density network with fast mobility, the change rate of topology is relatively high. However, topology changes would not be advertised until the update timer expires. Under such circumstances, topology changes might be too fast to be captured by periodic updates.

- **Resource Waste in Proactive Routing Protocols.** Proactive routing protocols propagate periodic topology updates irrespective of topology changes. However, in real-world scenarios, the nodes' mobility is more likely to be intermittent. Also, there might be only a fraction of the node population moving during a certain time period. Therefore, keeping a constant refresh rate may lead to unnecessary resource consumption.

To summarise, despite numerous performance studies on MANET routing protocols (see Chapter 3), it is still crucial to analyse the performance of signalling mechanisms in the presence of fre-
quent topology changes, and re-evaluate fundamental protocol design issues. Such analysis and re-evaluation are critical to understand how the signalling systems behave, from signalling functionalities in the network-layer protocols to signalling processes in the application-layer systems, and how to design effective signalling protocols for real-time services. The following paragraphs discuss the fundamental challenges of soft-state signalling in MANET routing protocols.

2.5 Fundamental Challenges

The challenges to efficient MANET signalling over networks composed of dynamic nodes and wireless links arise preliminarily because of the unique characteristics of MANET (i.e. topology dynamics and resource constraints) and network heterogeneity of wireless networks (i.e. diverse characteristics of the underlying wireless technologies). In particular, the following key issues have been identified in designing efficient signalling protocols for proactive MANET routing protocols:

- The quantitative relationship between signalling performance and various factors including refresh intervals and node velocity.

- Optimising signalling performance, i.e. improving throughput without introducing excessive control overhead, instead of balancing signalling performance between throughput and control overhead.

- Adaptive signalling in the presence of network heterogeneity. The volume of control messages should be determined by network conditions. The state description should be generic enough to provide additional flexibility in defining the objects carried by signalling messages.

- Incremental implementation and deployment of the proposed methods, which are expected to be generic and independent of specific routing protocols.

The following paragraphs give a further discussion on each issue.

- Quantifying Soft-state Performance. During the past decades, researchers from both the Internet community [1, 7, 30, 31] and wireless research teams [16, 24, 32] have advocated soft-state design as one fundamental protocol design principle. As has been discussed, some intuitive and high-level qualitative explanations have been provided on why soft state approaches have advantages over hard-state approaches. On the other hand, quantitative studies on signalling performance are limited. This might be explained by the simplicity and generality of soft state. Soft state mechanisms in IP networks was defined under the context of individual flows. However, the approach itself is very simple and generic, which is independent of application scenarios. This makes the modeling process difficult in terms of choosing performance metrics and identifying factors/variables, which are also expected to be generic. Moreover, the different scenarios of soft-state mechanisms, such as Internet and MANETs, have diverse characteristics and impact on signalling performance. Therefore, it is not feasible to propose a universal soft state model.
2.5. Fundamental Challenges

Traditional quantitative studies on soft-state performance are model-based and targeted at signalling in Internet scenarios [30, 33, 31, 34, 35]. For example, in order to evaluate the performance of soft state systems, Raman [30] built a queueing model using open-loop announce/listen process for data transport. Consistency and wasted bandwidth are used as the major metric, with loss rate and announcement death rate as the factors. Ji [33] compared the performance of soft-state and hard-state mechanisms with a continuous time Markov model. The impacts of loss rate, delay, retransmission timer and session length are examined on the performance of these two signalling approaches. Lui [31] evaluated the impact of the refresh timer period on robustness, using channel loss rate and a range of session characteristics as factors.

One drawback of these studies is that, the models are too generic to fit into a specific scenario. For example, data packet loss in MANETs can be caused by channel contention (channel loss rate), route unavailability and other reasons. Therefore, loss rate is too general as a factor in analysing MANET signalling performance. In addition, since the analysis is fully based on mathematic models, the accuracy of the analytical results largely depends on the applicability of the models/methods on the problem space (i.e. signalling performance). This is not fully validated in the previous studies.

The proposed approach to solving these problems is a combination of a model-based analytical method and a simulation-based performance evaluation method. Instead of proposing a general soft-state signalling model, the model in this study is based on a MANET routing process in presence of link dynamics. Correspondingly, state consistency is analysed under various factors, including node density, node velocity, refresh interval and radio range.

The focus on soft-state signalling under a MANET routing scenario facilitates further performance evaluation. This dissertation carries out simulation-based performance evaluations on the impact of refresh intervals under the same factors as in the analytic model. The observations from the simulations provide support for the results obtained from model-based analysis.

- **Optimising Signalling Performance.** As has been discussed in Section 2.4.2, the tussles between signalling performance and overhead are further intensified by the nature of MANETs (i.e. topology dynamics and resource constraints). Moreover, the requirement of scalability requires the routing protocols to reduce control overhead introduced by topology updates.

It is commonly believed that control overhead reduction by lowering the rate of topology updates would lead to performance degradation. The trade-off between performance and control overhead can be achieved by adjusting refresh intervals.

This dissertation adopts a step-by-step performance optimisation approach.

Firstly, this dissertation investigates the impacts of different soft-state updates on the routing (signalling) performance. This study finds that HELLO interval has a more significant impact on routing performance than topology update interval. From this we can infer that, lowering the rate of topology updates would not downgrade significantly the routing throughput, but could lead to
significant reduction of control overhead. Therefore, this finding can be used to optimise control overhead.

Secondly, this dissertation presents an in-depth study on various topology update strategies and their impacts on routing performance. The efficiency of topology advertisements was investigated from two major aspects, namely spatial redundancy (i.e. using extra topology state information in each topology control message) and temporal redundancy (i.e. adjusting the topology update frequency). The results from this study provide useful insights on how to effectively design flooding-based protocols and applications.

Finally, this dissertation proposes the Dynamic Timer algorithm and the Fast Neighbour Handshake algorithm (see Chapter 7). These two algorithms achieve better data packet delivery performance, while introducing much less control overheads than reducing HELLO intervals. Moreover, the Dynamic Timer algorithm could successfully mitigate the channel contention by dynamically adapting refresh intervals to network conditions.

- **Adapting Signalling to Network Conditions.** The use of various wireless devices, from handheld PDAs to vehicles, and the convergence of different networks make a real-world ad hoc network highly heterogeneous. Correspondingly, signalling protocols are required to adjust their behaviours and adapt themselves to underlying network conditions. An adaptive signalling approach requires the following properties for successful deployment:

  **Scalability.** The volume of control messages under the proposed algorithms should be bounded and not exceed that under the original routing algorithms. That is, adaptability should not be at the expense of scalability.

  **Automated Adaptability.** The adaptive algorithm should automatically adjust its behaviour to achieve target performance when operating in a wide range of network conditions. The dependency of its performance on parameter configurations should be reduced to the lowest level.

  **Efficient and Practical.** The adaptive algorithm should be self-contained. The dependency on other (unsolved) research issues, such as network measurement, should be avoided. Otherwise, the applicability of the algorithm might be jeopardised.

A typical existing adaptive routing protocol is SHARP [18], a hybrid routing algorithm which adopts optimal routing strategies based on separate application-level control requirements (i.e. minimising control packet overhead, controlling delay jitter and bounding loss rate). However, it resorts to extra techniques to measure network conditions (such as average link lifetime and average node degree) as well as traffic characteristics (such as loss rate and delay jitter).

The *Dynamic Timer* algorithm achieves adaptability to node movements through monitoring the dynamics of link change rate (of the node). Also, the approach is adaptive to traffic conditions through monitoring the (internal) data packet loss rate (of the node). Both are independent from
any specific proactive routing protocols or any extra techniques. Moreover, it also provides an extensible mechanism that allows using extra techniques to add new adaptability such as to channel capacity and channel status (i.e. congestion etc).

- **Incremental Deployment.** Soft state mechanisms have been widely adopted in a variety of protocols and applications, including proactive MANET routing protocols. In order to facilitate incremental implementation and deployment, the proposed improvements on soft-state signalling are also expected to be *generic* and *independent* of any specific proactive MANET routing protocols.

The *generic* aspect of the proposals can be explained as follows.

- The analytical models proposed are generic. The soft-state model in Chapter 5, the topology update model in Chapter 6 and the neighbour detection model in Chapter 7 are rooted on the basic soft-state mechanism, which have no dependency on any MANET routing protocols.

- The signalling algorithms are independent of any specific proactive routing protocols or any other network technologies (such as network measurements). In general, the algorithms can be applied to the protocols or services where soft-state signalling has been used. For example, the proposed neighbour detection algorithms in Chapter 7 are not limited to OLSR and DSDV, since most of the existing proactive MANET routing protocols provide mechanisms to calculate statistics of link change events.

- The implementations are modularised. Interfaces are defined in implementing the signalling options in topology update strategies (see Chapter 6) and neighbour detection algorithms (see Chapter 7). In this way, new signalling methods can be implemented with no need to revise the core routing implementations. Such modular design also facilitates porting the signalling implementations between different routing protocols.
Chapter 3

Background and Related Work

In this chapter we introduce the reader to background information on signalling over mobile wireless networks. We start with a description of the basic signalling concepts of network signalling. Especially we survey some key concepts on generic soft-state signalling in Section 3.1.3. In Section 3.2, we illustrate the detailed signalling mechanisms used in the most popular MANET routing protocols in terms of neighbour detection and topology advertisement. We follow this with a discussion of recently proposed enhancements to MANET signalling in Section 3.3, including several recent studies on adaptive routing approaches, highlighting their weaknesses. We conclude this chapter with a summary in Section 3.5.

3.1 Network Signalling

Network Signalling is defined as the exchange of control information among elements of a telecommunication network to initiate, maintain, and release connections and for network management [36]. A signalling protocol is a set of message definitions and state machines in the cooperating entities involved in the signalling message processing.

In this section, we first provide an overview of the fundamental signalling concepts. Then we present state-of-art studies on soft-state signalling.

3.1.1 Signalling Concepts

Network signalling is about network control. It is to install and maintain state by exchanging messages between the nodes, in order to keep states consistent and remove obsolete states [29].

Network signalling can be related to specific flows, or to network operation in general. It can involve nodes in the network, or run transparently between the end hosts.

If the management of network control state is at the level of a specific flow, the signalling is defined as per-flow signalling.¹ Signalling protocols of this type carry per-flow information for the applications, such as resource reservation information in RSVP-like QoS signalling protocols. Signalling messages may install, modify, refresh, or simply read per-flow state information stored in network elements for particular data flows. Usually a network element also manages this information at the per-flow level.

In contrast, if the management of network control state is at the level of network operations, instead

¹Formally, a flow is a sequence of packets that are semantically related, and a session can be defined as a flow with a specific destination address and transport protocol.
of a specific data flow, the signalling is defined as *per-node signalling*. Routing protocols, for example, exchange control messages in discovering and maintaining connections between two end nodes. In this study we focus on this type of signalling, instead of per-flow signalling.

From the aspect of message delivery methods, signalling can be classified into two types: in-band signalling and out-of-band signalling. In-band signalling is sending data and control information in the same channel. In contrast, out-of-band signalling is the exchange of control information in a separate channel of data, or sending control packets and data packets separately. This study focuses on out-of-banding signalling, as used in most MANET routing protocols.

From the aspect of participating entities, network signalling can be classified into user-network, network-network, and user-user signalling. The User-Network signalling concerns control messages passed between user equipment and a network node. Network-Network signalling is the control signals among network nodes. The User-User signalling is the control signals between end users. MANETs blur the distinction between user devices and network nodes, therefore such signalling difference does not exist.

### 3.1.2 Signalling in Wireless Networks

There have been several signalling approaches designed for wireless networks. For example, the INSIGNIA signalling system [32] is a lightweight IP-based QoS (Quality of Service) framework that supports end-to-end resource reservation. INSIGNIA uses in-band signalling to reduce bandwidth consumption of signalling traffic. It uses soft-state *time-out* approach to maintain resource reservations. In particular, each flow is associated with a soft-state timer. The soft-state timer is continually refreshed as data packets associated with a flow are received at intermediate routers. If data packets are not received within a period, reservation state times out and the resource is deallocated.

A performance evaluation of INSIGNIA is presented in [37], including the impact of soft-state time-out intervals on network performance. The results show that, as the timer interval increases, the "ability to support adaptive services" of INSIGNIA decreases. Especially, *false restoration* is observed when the time-out interval is smaller than 2s. In order to remove resource lock-ups and false-restorations, an adaptive soft-state timeout approach is proposed to adjust the *timeout* intervals based on the measured data packet inter-arrival rate.

Based on their behaviours in state installation/maintenance, existing signalling approaches in wireless networks can be classified into two basic types: *reactive signalling* and *proactive signalling*.

- **Reactive signalling** initiates and stops message exchange as requested. For example, route maintenance via HELLO messages in AODV [25, 38] is in fact a reactive signalling process, since HELLO messages are sent only when there is data transfer to the destination nodes. The route state information is maintained only during the data transfer period. After the data transfer, no HELLO messages are exchanged between any neighbouring nodes in the routes.

Reactive signalling is usually divided into three stages: signalling initiation, signalling maintenance and signalling termination. Therefore, messages in reactive signalling include two types,

---

2In other studies, they are called path-coupled signalling and path-decoupled signalling
trigger messages that install/remove states, and refresh messages that keep state updated.

- **Proactive signalling** exchanges signalling messages periodically, based on soft-state mechanisms. The signalling process is not correlated with data forwarding (or data flows). For example, the topology advertisement process in most proactive MANET routing protocols is a proactive signalling process. Topology messages are advertised periodically in the network.

Proactive signalling usually does not send separate trigger messages to install state. Instead, it just inserts new state into signalling messages. The receivers of such messages check each state update message against their state repositories. New state is installed while existing state is simply refreshed. Similarly, proactive signalling does not send explicit state removal messages. Instead, the obsolete state is timed out if unrefreshed within a certain period.

### 3.1.3 Soft State Signalling

In this section, we survey some key concepts on generic soft-state signalling, including analytical results on soft-state performance and scalable timer approaches.

**Overview**

The concept of soft state was first introduced by Clark [7] in the late 1980s, as a fundamental signalling mechanism in maintaining *flow state* information for Internet services. Through exchanging periodic messages, the end nodes restore flow state information that is lost in a node failure and enforce the services associated with the flows. The reason for the wide use and deployment of this design principle is that it works well in a variety of Internet protocols in practice. In particular, soft state has the following characteristics:

- **Robustness.** Soft state affords "survivability in the face of failure" [7]. Systems built on soft state are self-healing, since loss of session states can be recovered automatically and would not result in permanent service outage. Soft state protocols use the timeout mechanism to create a virtual and predictable feedback channel, by which loss of signalling messages would not generate orphaned state.

- **Predictable Control Overhead.** The control overhead of soft state signalling is determined by the number of network states and nodes, the degree of connectivity of each node, and refresh intervals. The frequency of message exchange does not change with network conditions. Therefore, soft state protocols are less likely to congest the network with signalling traffic when network conditions deviate from the norm.

- **Simplicity in Implementation.** A soft state mechanism does not depend on other techniques, such as network measurement or movement detection. It can be deployed incrementally without any need to change existing system implementation.

On the other hand, the soft state approach has several drawbacks. Soft state itself does not provide adaptive procedures or mechanisms to change refresh frequency when the topology or activity of the
network changes. Therefore, there is no performance guarantee in the presence of network dynamics, and the control overhead is not optimised, which might bring scalability problems and lead to resource waste. In addition, the refresh frequency typically has to be configured manually, which increases complexity in system maintenance especially under certain performance requirements.

Theoretical Performance Analysis

Raman [30] develops an open-loop multi-class queuing model for soft-state based communication. The transmission channel between sender and receiver acts as a *service*. Consistent state and inconsistent state are inputs of the queuing system. Using this model, they conduct queuing analysis to characterise data consistency and performance tradeoffs under a range of work loads, network loss rates and session expiration rates. Then they extend the model with feedback and show through simulation that adding feedback dramatically improves data consistency without increasing network resource consumption. Finally a Soft State Transport Protocol is proposed based on probabilistic delivery.

Ping et al [33] carries out a theoretical performance analysis of several hard-state and soft-state protocols, including a pure soft-state approach, soft-state approaches with explicit state removal and reliable message delivery, and a pure hard-state approach. A continuous time Markov model is developed to quantify the signalling performance in terms of consistency and cost in single and multi-hop signalling scenarios. The study finds that, applying explicit state removal into the soft-state approach could substantially improve signalling performance while introducing little extra traffic overhead. In addition, applying reliable control message delivery mechanism into the soft-state approach could outperform the hard-state approach.

Lui et al [31] investigates the *robustness* of the soft state signalling approach. This study develops simple models to compare the performance of the soft-state approach with the hard-state approach under the following three circumstances: (1) Denial of Service (DoS) attacks, (2) overloaded (lossy) communication channels and (3) broadcast sessions with dynamic participants. The study concludes that, the hard-state approach can be configured to outperform the soft-state approach, if network conditions are predictable. And the soft-state approach is more resilient if network conditions are unknown.

Scalable Timer

*Per-session* based soft state signalling, such as flow state maintenance in RSVP, has the scalability problem in terms of the amount of state. For example, as the size and usage (i.e. the number of flows) of the network grow, the amount of state to be maintained also increase. With the traditional fixed-timer approach, the control traffic grows linearly with the increase of the amount of flow state, since separate refresh messages are sent for each flow state.

In order to tackle this problem, Sharma et al proposes a *scalable timer* approach to limit control traffic overhead [39]. In this study, the refresh interval is adjusted according to the fixed available bandwidth (allocated for control traffic) and the amount of state to be refreshed. In particular, the refresh messages are sent in a round robin manner in order to adapt the bandwidth consumption to the fluctuations in the amount of network state over time. The priorities of message delivery are given to trigger messages in order to improve the response time. In order to determine the value of the time-out timer intervals, the
3.2. Routing Protocols for Mobile Ad Hoc Networks

In this section, we present a quick overview of routing protocols for MANETs. Instead of listing all of the details of routing operations, our discussion is focused on the signalling mechanisms in routing state maintenance, including neighbour sensing and topology advertisement.

3.2.1 Traditional Proactive Routing: LS vs. DV

In Link State (LS) protocols like OLSR [16], each node discovers and maintains a complete and consistent view of the network topology, by which each node computes a least-cost path tree with itself as the root, and applies the results to build its forwarding table. This assures that data packets are forwarded along the least-cost paths to their destinations. Note that the cost metric is hop count.

LS protocols rely on periodic refresh messages to reflect topology changes and maintain correct topology information. Each node sends HELLO messages periodically to discover new neighbours and detect link failures. Unlike LS protocols such as OSPF, in which the topology update is triggered by network change events, LS protocols in MANETs advocate periodic topology update strategy, in order to avoid the large amount of topology update messages triggered by frequent topology change events.

In Distance Vector (DV) protocols like DSDV [23], each node maintains a routing table containing the distance from itself to all other nodes in the network. Each node broadcasts periodically its routing table to each of its neighbours and uses similar routing tables from neighbouring nodes to update its table. The route selection is based on the Distributed Bellman-Ford (DBF) algorithm. To keep up with network changes, DV algorithms use both periodic and triggered updates.

The main problem of traditional proactive routing (especially the LS algorithm) lies in the use of fixed timer intervals. The refresh intervals are configured by administrators, usually with the default values recommended by protocol designers. Basically, high mobility demands small intervals to speedup link change detection, while low mobility only needs relatively large intervals to reduce control overhead. Due to the non-uniform distribution of node mobility, both temporally and spatially, the fixed timer intervals fail to be effective when/where the node mobility is high and efficient when/where the node mobility is low. Thus, the refresh intervals need to be adapted to network conditions.
3.2. Routing Protocols for Mobile Ad Hoc Networks

3.2.2 Proactive Routing

OLSR

In this section, we summarise the recent research efforts on Optimised Link State Routing (OLSR) protocol [16], including performance analysis and extensions.

- **Overview.** OLSR is a table-driven proactive routing protocol for MANETs.

  OLSR inherits the concept of the link state algorithm and uses the Shortest Path First (SPF) algorithm. It exchanges topology information with other nodes of the network periodically. Each node maintains the topology information for the whole network, based on which the routes are computed. Due to its proactive nature, it has the advantage of making the routes ready before they are needed.

  On the other hand, OLSR inherits the concept of forwarding and relaying from HIPERLAN. It uses MPR (Multi-Point Relay) [40, 41, 42] to optimise flooding. Each node selects a set of its neighbour nodes as MPRs. And only MPR nodes are responsible for forwarding control traffic. MPR provides an efficient mechanism for flooding control traffic by reducing the number of transmissions required. More details on MPR selections and routes calculations algorithms could be found in RFC 3626 [19].

- **State Repository.** In order to maintain the topology information of the whole network in presence of mobility and failure, OLSR daemon needs to record and keep updated the following state information in its internal tables.

  *Link tuples* in link set keep track of the link status between the node and its neighbours. There are two types of status: *SYM* link (e.g. bi-directional) and *ASYM* link (e.g. single-directional). Each link tuple contains the interface addresses of the local node and the neighbour node (e.g. the end points of a link), and valid time until which the link is considered as valid link (otherwise link must be removed), ASYM link and SYM link.

  *Neighbour set* contains neighbour tuples to keep track of a node's neighbour status, including willingness and valid time etc, while 2-hop neighbour set records a set of 2-hop tuples that describe symmetric links between its neighbours and the symmetric 2-hop neighbourhood.

  *MPR set* maintains a set of neighbours that are selected as MPR, while MPR selector set records a set of MPR-selector tuples and describes the neighbours that has been selected this node as a MPR.

  *Topology set* maintains topology information about the network. This information is acquired from Topology Control (TC) messages and is used for routing table calculations.

- **Consistency Issue.** Basically, the state management mechanism of OLSR consists of two stages. First, neighbour-sensing process detects new neighbours (node joining) by periodically broadcasting HELLO messages; or, timer mechanism of internal tuples detects node leaving or link-broken
3.2. Routing Protocols for Mobile Ad Hoc Networks

Topological consistency is an important property of a routing protocol. The consistency problem also consists of two parts: the consistency between the internal tables in one node and the consistency between tables in different nodes. Internal consistency could be easily achieved by careful table maintenance operation. Neighbour sensing process updates link set. Upon receiving a HELLO message, OLSR updates link set, neighbour set and 2-hop neighbour set, and re-calculate MPR set and MPR selector set. Upon receiving a TC message, OLSR updates its topology information. Route table will be re-computed once the node receives any OLSR messages. If link tuple timer expires, link set, neighbour set and 2-hop neighbour set will be updated; MPR set and routing table will be re-computed.

On the other hand, it is not so easy for OLSR to maintain external consistency. When there is a failure in the network (link or node failure), the protocol cannot detect the failure until the related timer expires. Further, the failure information will not be propagated until the TC timer expires. Moreover, a consistent view of the new topology will not be re-established in network wide until each node has successfully received the TC messages; loss of such a TC message with failure information might cause another delay of one TC interval. During such a transient, the data traffic forwarded toward a failed node will be dropped. Even worse, routing loops may emerge because of inconsistency in routing tables and may lead to congestion in the network.

- **State Maintenance.** OLSR simply relies on the soft-state approach to maintain the consistency of topology information among nodes in the network. Apart from the periodic control messages, OLSR does not generate extra control traffic in response to link failure and nodes joining/leaving events.

In OLSR, two types of control messages are used for state maintenance: the HELLO message and the Topology Control (TC) message. A node broadcasts periodic HELLO messages to the other nodes within its radio transmission range. When a new node joins the network, the node detects its new neighbours when receiving the first HELLO message. When nodes leave the network, or the links between them get broken, the corresponding link state and the neighbour state is removed after the state timeout timers expire without receiving any HELLO messages. TC messages are sent and relayed by MPR nodes [19]. TC messages sent by node A contain a list of the neighbouring nodes that have selected A as their MPR. TC messages enable the remote nodes to discover the links between A and its selectors. Based on such information, the routing table is calculated based on the SPF algorithm. In addition, periodic TC messages help the remote nodes recover from loss of topology information caused by state inconsistencies or node restarts.

Each OLSR node has four timers associated with the control traffic. HELLO refresh timer is used to clock out the HELLO messages to refresh existing state and advertise new neighbour state. TC refresh timer is used to clock out the TC messages to refresh existing topology state and advertise topology changes. Neighbour state timeout timer is to remove a neighbour state entry if the node
does not receive a HELLO message for that state before the state timeout timer expires. Similarly, the topology state timeout timer is used to remove a topology state entry if the node does not receive a TC refresh message for that state before the state timeout timer expires.

By default, the neighbour holding time (i.e., the interval of the neighbour state timeout timer) is set to be 3 times the value of the HELLO interval. The topology information holding time (i.e., the interval of the topology state timeout timer) is 3 times the default value of the TC interval.3

- Overhead Reduction Techniques. Compared with a pure link state protocol, the major contribution of OLSR is its optimization in reducing flooding messages as control traffic overhead. In a pure link state protocol, all the links with neighbour nodes are declared and are flooded in the entire network. OLSR optimizes the pure link state protocol for mobile ad hoc networks with MPR technique, which minimizes the flooding of broadcast control packets in the network by reducing duplicate retransmissions in the same region. The protocol uses the MPRs to facilitate efficient flooding of control messages in the network.

First, it reduces the size of control packets. Instead of all links, it declares only a subset of links with its neighbours who are its multipoint relay selectors.

Secondly, it minimizes flooding of this control traffic by using only the selected nodes, called multipoint relays (MPR) [19], to diffuse its message in the network. Only the multipoint relays of a node retransmit its broadcast message. Such a technique significantly reduces the number of retransmissions in a flooding or broadcast procedure.

- OLSR Performance Analysis. Jacquet and Viennot [43] give an analysis on control overhead of MANET routing protocols. In this study, the reactive protocols are called “flooding protocols”, since their route discovery is based on flooding control packets, while the proactive protocols are called “HELLO protocols”, since periodical HELLO messages are emitted to detect the topology changes. The study concludes that, flooding protocols outperform HELLO protocols when the node mobility is high and the number of active routes is small. And HELLO protocols are more suitable when the number of active routes are large.

Jacquet et al focus on the MPR concept in [40, 41] and present a theoretical performance analysis under two network models: the random graph model and the unit graph model. The performance metrics include the average MPR size and the retransmission number. The performance of OLSR, featured with optimized flooding based on MPR, is compared with the performance of other link state routing protocols using “dominating set flooding”.

INRIA gives its simulation results on OLSR performance in [44]. The key metrics in this study include control overhead, loss ratio of control packets and average route availability. The results in small networks show that, although MPR mechanism optimizes control overhead, it leads to worse average route availability. When there is less data traffic load or more available network resources, the performance without MPR is slightly better than that with MPR. MPR offers fair

3Value 3 is derived from RFC 3626 [19]
performances at average load and better performance at high load. The results in the large network are different. OLSR with MPR performs better than that without MPR.

- **Tuning OLSR parameters.** Gomez et al [45] analyses the impact of routing protocol parameter settings on OLSR routing performance. It studies the effects of tuning OLSR configuration (e.g. HELLO Interval and TC interval) on RCL (Route Change Latency), which is defined as the re-routing latency after a link failure; further efforts explore the effect of the measured RCL values on end-to-end path connectivity using a simple end-to-end connectivity model; finally the study concludes that end-to-end connectivity can be enhanced using different parameter settings from the default ones.

There are some flaws in this study. Firstly, the measurement of RCL is coarse since other factors may also contribute to the connectivity gap, besides the re-routing transience. Secondly, assumptions made in the end-to-end connectivity model are not reasonable. For example, this study assumes the network never gets partitioned and the path length remains unchanged after the re-routing. Essentially, the end-to-end connectivity model is not realistic with the above assumptions under the scenario of MANETs.

- **QoS Extensions to OLSR.** QOLSR [46] is an extension to OLSR, which provides optimal QoS routing functionalities in terms of available bandwidth and delay. QOLSR inherits OLSR's MPR technique to minimise control traffic. This technique significantly reduces the number of retransmissions required to flood a message to all nodes in the network. QOLSR requires only partial link state and their QoS information to be flooded in order to provide optimal routes under QoS constraints as those in the whole network topology. As in OLSR, all nodes, selected as MPRs, MUST declare the links QoS information to their MPR selectors using TC messages.

Compared with OLSR, QOLSR has the following improvements in order to provide optimal routes under QoS constraints as those in whole network topology. In neighbour sensing, each node estimates the QoS conditions (available bandwidth, delay, loss rate, power consumption, etc.) on links to each neighbour. In topology advertisement, QoS conditions of the links between the node and its MPR selectors are added into each TC message. The routing table is calculated with a shortest-widest path algorithm (a variant of the Dijkstra routing algorithm) to find a path with maximum bandwidth.

**DSDV**

The Destination Sequenced Distance-Vector Routing protocol (DSDV) [23] is a table-driven proactive routing algorithm based on the classical Bellman-Ford routing mechanism. Each node in the network maintains a routing table that records distance vectors, i.e. the number of hops to all of the possible destinations within the network and the corresponding next-hop nodes.

The main improvement made to the Bell-Ford algorithm is the loop-free property by marking nodes with sequence numbers, which are used to distinguish stale routes from new ones. The route labeled with
3.2. Routing Protocols for Mobile Ad Hoc Networks

the most recent sequence number is always used. In the event that two updates have the same sequence number, the route with the smaller metric (such as number of hops) is used in order to optimise the path.

DSDV nodes keep track of the settling time of routes for damping fluctuations, calculated by "maintaining a running, weighted average over the most recent updates of the routes, for each destination". By delaying the routing update broadcast for a settling period, nodes reduce network traffic and optimise routes by avoiding broadcasts that might be superseded by a better route in the very near future.

- **Topology Update Strategies.** DSDV requires each node to advertise its own routing table by broadcasting its entries to each of its current neighbours *locally*. In order to reduce the amount of state information carried in each update and help alleviate the potentially large amount of topology update traffic, DSDV employs two types of update packets.

  - **Full dump updates** carry all available routing information and might require multiple network protocol data units (NPDUs). Full dumps can be transmitted relatively infrequently when no movement of mobile nodes occurs.

  - **Incremental dumps** carry only information changed since the last full dump. The size of incremental dumps is smaller than that of full dumps, and therefore can fit into a standard NPDU. When movement becomes frequent, the size of incremental dump increases and approaches the size of a NPDU. Then a full dump can be scheduled so that the next incremental will be smaller.

- **Neighbour Sensing Schemes.** Clearly, new neighbours can be detected by exchanging periodically the routing tables. There are two proposals in link breakage detection, either by the layer-2 protocol, or by time-out if no routing table updates have been received for a period from an existing neighbour. When a link to a next hop is broken, any route through that next hop is immediately assigned a metric of $\infty$ so that it should not be selected for data delivery.

**TBRPF**

The Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) [24] protocol is a link-state routing protocol designed for MANETs, which provides hop-by-hop routing along shortest paths to each destination. Each TBRPF node computes a source tree based on partial topology information stored in its topology table.

- **Topology Update Strategies.** Unlike flooding mechanism in OLSR, link state update in TBRPF is broadcasted along a dynamic min-hop-path tree rooted at the update source node. The update stops at the leaf nodes of the tree. The broadcast tree is maintained with the topology information received by each node in the tree. Sequence numbers are used to ensure freshness of the topology updates.

Unlike a typical proactive routing protocol, TBRPF does not send link-state updates periodically. Instead, the updates are sent in reaction to link changes.
3.2. Routing Protocols for Mobile Ad Hoc Networks

- **Neighbour Sensing.** HELLO messages are used by TBRPF in neighbour detection. TBRPF uses a combination of periodic and differential updates to keep all neighbours informed. TBRPF performs neighbour discovery using differential HELLO messages that report only changes in the status of neighbours. Each node also has the option to report additional topology information (up to the full topology), to provide improved robustness in highly mobile networks.

**WRP**

Wireless Routing Protocol (WRP) described in [47] is a table-based protocol maintaining routing information among all nodes in the network. Each node in the network is responsible for maintaining four tables: distance table, routing table, link-cost table and message retransmission list table (MRL).

Each entry of the MRL contains the sequence number of the update message, a retransmission counter, an acknowledgment-required flag vector with one entry per neighbour, and a list of updates sent in the update message. The MRL records which updates in an update message need to be retransmitted and which neighbours should acknowledge the retransmission.

WRP nodes are loop-free by exchanging the distance and second-to-last hop information for each destination. It avoids the count-to-infinity problem by forcing each node to perform consistency checks of predecessor information reported by all its neighbours. This ultimately eliminates looping situations and provides faster route convergence when a link failure event occurs.

- **Topology Update Strategies.** WRP nodes send update messages only between neighbours. Each update message contains a list of updates (the destination, the distance to the destination, and the predecessor of the destination), as well as a list of responses indicating which mobiles should acknowledge (ACK) the update. Update messages are sent after the nodes process updates from neighbours or detect link changes, such as link breakage. The neighbours modify their distance table entries and check for new possible paths through other nodes. Any new paths are relayed back to the original nodes so that they can update their tables accordingly.

- **Neighbour Sensing.** WRP nodes discover the existence of their neighbours through receiving acknowledgments and other messages. If a node is not sending messages, HELLO messages are broadcasted. The lack of messages from the node indicates the failure of that link. When a node A receives a HELLO message from an unknown node B, node B is added to node A's routing table.

### 3.2.3 Reactive Protocols

**AODV**

The Ad Hoc On-Demand Distance Vector (AODV) protocol [25, 38] is a typical on-demand (or reactive) routing algorithm designed for MANETs. It discovers and maintains routes between nodes only when requested by source nodes of data traffic. In addition, AODV supports multicast routing by building trees that connect multicast group members. AODV uses sequence numbers to ensure the freshness of routes. It is loop-free, self-initiated and "scales to large numbers of mobile nodes".

- **Route Discovery & Maintenance.** AODV relies on route request query (RREQ) and route
reply (RREP) in route discovery. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet with (current & the most recent) sequence numbers for the destination across the network. Nodes receiving this packet update their route information for the source node and set up backwards pointers to the source node in the route tables. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. A RREP packet is sent back to the source by unicasting. Otherwise, it re-broadcasts the RREQ query. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ that they have already processed, the RREQ is discarded and will not be forwarded.

The routes are maintained as long as they remain active. A route is considered active as long as there are data packets periodically traveling from the source to the destination along that path. Once the source stops sending data packets, the links will time out and eventually be deleted from the intermediate node routing tables. If a link break occurs while the route is active, the node upstream of the break propagates a route error (RERR) message to the source node to inform it of the now unreachable destination(s). After receiving the RERR, if the source node still desires the route, it can reinitiate route discovery.

- **Neighbour Sensing.** In order to maintaining active routes, AODV has two methods in detecting link breakage: using HELLO messages and link layer notification (or passive acknowledgment). AODV nodes of active routes exchange periodic HELLO messages with their neighbours. If a node receives no HELLO messages within a period from one of its neighbours, or is notified by link layer that data packet delivery to one of its neighbours has failed, or receives no acknowledgment from one of its neighbours, the link to the neighbour is assumed to be lost.

AODV does not need mechanisms to detect neighbour arrivals, since after discovered, the established routes remain unchanged even if a shorter path appears due to link establishment.

**DSR**

The Dynamic Source Routing (DSR) protocol [48] is an on-demand routing protocol based on the concept of *source routing*. DSR nodes are required to maintain route caches that contain the source routes. Entries in the route cache are continually updated.

When a source node desires a route to a destination, it first consults its route cache to determine whether it already has a route to the destination. If not, it initiates route discovery by broadcasting a route request packet. This route request contains the address of the destination, along with the source node's address and a unique identification number.

Each node receiving the route request packet checks whether it has a route to the destination. If it does not, it adds its own address to the route record of the packet and then forwards the packet along its outgoing links. To limit the number of route request retransmission, each node keeps a record of the route requests.
A route reply is generated when the route request reaches either the destination itself, or an interme-
diate node that contains in its route cache an unexpired route to the destination. When the packet reaches
either the destination or such an intermediate node, it contains a route record yielding the sequence of
hops taken. To send the route reply, the responding node must have a route to the initiator. If not, the
node may reverse the route in the route record (as long as symmetric links are supported). Otherwise,
the responding node may initiate its own route discovery and piggyback the route reply on the new route
request.

Route maintenance of DSR is through the use of route error packets and acknowledgments. Route
error packets are generated at a node when the data link layer encounters packet transmission failure.
When a route error packet is received, the corresponding route is removed from the node’s route cache.
In addition to route error messages, acknowledgments are used to verify the correct operation of the
route links. Such acknowledgments include passive acknowledgments, where a mobile is able to hear
the next hop forwarding the packet along the route.

TORA

The Temporally Ordered Routing Algorithm (TORA) [26] is a highly adaptive loop-free distributed
routing algorithm based on the concept of link reversal. It is source-initiated and provides multiple
routes for any desired source/destination pair.

The key design concept of TORA is the localisation of control messages to a very small set of
nodes near the occurrence of a topological change. To accomplish this, nodes need to maintain routing
information about neighbouring nodes.

During the route discovery and maintenance phases, nodes use a height metric to establish a directed
acyclic graph (DAG) rooted at the destination. Thereafter, links are assigned a direction (upstream or
downstream) based on the relative height metric of neighbouring nodes. If the DAG route is broken, a
DAG rooted at the same destination is re-established. Upon failure of the last downstream link, a node
generates a new reference level, which results in the propagation of that reference level by neighbouring
nodes, effectively coordinating a structured reaction to the failure. Links are reversed to reflect the
change in adapting to the new reference level. This has the same effect as reversing the direction of one
or more links when a node has no downstream links.

The height metric is dependent on the logical time of a link failure. TORA assumes that all nodes
have synchronised clocks (accomplished via an external time source such as the Global Positioning
System). A new reference level is defined each time a node loses its last downstream link due to a link
failure.

TORA needs notification support from link layer or alternative techniques in monitoring link status
changes. It eliminates invalid routes by flooding a broadcast clear packet (CLR) throughout the network.

One potential problem of TORA is oscillations, especially when multiple sets of coordinating nodes
are concurrently detecting partitions, erasing routes, and building new routes based on each other. The
oscillations are expected to be temporary and route convergence will ultimately occur.
3.2. Characterisation of MANET Routing Dynamics

There have been several attempts to characterise link dynamics and route dynamics of MANETs.

Johansson et al proposed the relative motion between mobile nodes to distinguish the different mobility models used for their scenario-based study in [49]. Empirical distributions of link lifetime have been presented in [50] under various simulation parameters. Prince et al [51] developed a mathematical model on the statistics of link dynamics and derived analytical expressions for a number of link properties, including expected link lifetime, expected link arrival rate, expected link breakage rate, expected link change rate and link change interval time; the study concluded that the link change inter-arrival time density can be modeled by an exponential function.

Sadagopan et al [52] uses average relative speed to analyze the probability distribution of link duration and path duration in reactive MANET routing protocols, under various mobility models including the Random Waypoint, Reference Point Group Mobility, Freeway and Manhattan model; they discover that, at moderate and high velocities the exponential distribution is a good approximation of the path duration for a range of mobility models.

We investigated route duration statistics in proactive MANETs, as shown in Appendix B. An analytical approach is developed to obtain the network statistics of link and path durations including probability density functions (PDFs). We find that, in a mobile ad hoc network with moderate or high mobility, the route duration could be approximated by an exponential distribution with appropriate parameters.

3.2.5 Performance Comparison of Routing Protocols

During the past decade, there have been a large number of performance evaluation studies on MANET routing protocols [49, 53, 54, 56, 57, 58, 59, 60, 61]. Here we pick two real-world experimental efforts, which are closely related to this research.

Choi and Ko [56] evaluate the performance of four well-known ad hoc routing protocols: AODV, DSR, LAR and OLSR, using QualNet under realistic military scenarios. The results show that the DSR and LAR protocols achieve relatively good performance, compared with OLSR and AODV. More specifically, OLSR performs the worst among the routing protocols on packet delivery ratio in terms of node velocity and network density. This shows OLSR's incapacity in reacting to topology changes and control traffic congestion. Surprisingly, OLSR has relatively bad performance on average end-to-end delay compared with LAR and DSR in terms of node velocity. Moreover, its performance on average delay is the worst with increasing node density. As a table-driven routing protocol, OLSR shows the worst performance on overhead, which is as expected due to the periodic refresh control messages.

Hsu et al [55] presents a similar study on the performance of AODV, DSR, OLSR, OSPF and ZRP under realistic network scenarios. It is based on an actual exercise carried out under the DARPA PCS communications program. The results show that, OLSR has a uniform but fairly sharp decline on throughput with the increase of network density. OLSR has the largest latency and the highest slope, which indicates that OLSR has difficulty in scaling to the number of hops.
3.3 Signalling in Proactive Routing Protocols

In this section, we give further details about some signalling mechanisms in proactive MANET routing protocols.

3.3.1 HELLO Based Neighbour Detection

MANET routing protocols, such as OLSR [16] and AODV [25], detect neighbour changes in a proactive manner: periodic exchange of HELLO messages. A generic description of the proactive neighbour detection approach could be found in the Internet draft [62]. Here we only give a brief introduction of the neighbour discovery process.

1. When a HELLO message from node \( B \) is received by node \( A \), node \( A \) check in its link set if the message is from an existing neighbour. If not, a new link entry \((B \to A)\) is created and marked as asymmetric; otherwise, node \( A \) simply updates the link status.

2. Node \( A \) broadcasts a HELLO message, containing \( B \) in the list of asymmetric neighbours. When receiving this message, \( B \) infers the asymmetric link \((B \to A)\). Thus, since it receives HELLO messages from \( A \), \( B \) concludes that the link between \( A \) and \( B \) is symmetric.

3. Node \( B \) broadcasts a HELLO message and advertises node \( A \) as a \( SYM \) neighbour. Node \( A \) receives this message and marks the asymmetric link entry \((B \to A)\) as symmetric \((B \leftrightarrow A)\).

4. Periodic exchange of HELLO messages enables the nodes to keep track of the status of the neighbouring nodes.

5. If, within a certain period, the node \( A \) does not receive a HELLO message from \( B \), \( A \) assumes that the link to this neighbour has been lost.

In order to avoid synchronisation of the control packets (i.e. several neighbour nodes attempt to transmit control traffic simultaneously), each node simply adds a randomly generated jitter to the interval at which HELLO messages are generated.

Because each node sends HELLO messages periodically, the protocol can tolerate reasonable message loss without requiring reliable transmission. Such losses can occur frequently in radio networks due to collisions or other transmission problems.

3.3.2 Cross-layer Neighbour Detection

In addition to traditional HELLO based neighbour sensing, link layer notification can be used to speed up link change detection.

RFC 3626 [19] suggests that link layer notification could facilitate neighbour detection. If the link layer information is available for the routing layer, which describes connectivity to neighbouring nodes, this information can be used to maintain neighbour status, as supplements to HELLO based neighbour sensing scheme. For example, Voorhaen and Blondia [20] use data packet delivery failure events as link layer feedback to help link breakage detection. Such schemes introduce no extra control overhead, but request support from the MAC layer implementation.
3.3.3 Topology Update Strategies

Samar and Haas [63] propose several update strategies that calculate the update periods under certain requirements (such as bounded-delay) and assumptions on the distribution of link change arrival. The basic motivation is to maximise the update period while maintaining the performance. The simulation results show that the proposed schemes, while keeping satisfactory network performance (e.g. throughput), can lead to more than 45% control traffic reduction compared with traditional OLSR strategies.

Clausen applies the concept of fish eye [64] into OLSR [65] and proves the control overhead reduction by introducing temporal partiality into proactive updates through simulations.

3.3.4 Adaptive Signalling

In the following paragraphs, we list several signalling approaches in improving adaptability of MANET routing protocols.

FAST OLSR

In order to meet the need for fast mobility in Mobile Ad-hoc Networks, Benzaid et al [17] presented a FAST-OLSR extension to the Optimised Link State Routing protocol (OLSR). A fast moving node refreshes the links to its MPR nodes at a higher frequency than its non-MPR neighbours by means of Fast-HELLOs. The Fast-HELLO messages only contain the address of its MPRs. Fast-OLSR extension aims at reducing data packet loss rate while keeping the overhead reasonable.

The proposed Fast-OLSR extension tries to maintain a partial topology connectivity of MPR links. However, due to the delay in establishing MPR relationship with new neighbours, a smaller HELLO interval might not improve neighbour discovery. In addition, if there is no alternative route for a broken MPR link, Fast-OLSR would not show any advantages over standard OLSR, since the latency of new MPR links establishment is the same (i.e. non-MPR links are refreshed at normal rate). Most importantly, it is impossible to judge the movement speed only based on the change rate in its neighbours. When a node starts to move, the change rate in both itself and each of its neighbour might increase. Because of the variance in radio ranges, the change rate of the node itself may be larger than its neighbours. Therefore, the applicability of the proposed Fast-OLSR algorithm is questionable.

Hybrid Routing Protocols: SHARP

Ramasubramanian et al [18] propose a hybrid routing algorithm that adopts optimal routing strategies, both proactive and reactive, based on separate application-level control requirements (i.e. minimising control packet overhead, controlling delay jitter and bounding loss rate). It does this by defining proactive zones around some nodes. The nodes at a distance less than or equal to the zone radius are within the proactive zone and maintain routes proactively only to the central node. All nodes not in the proactive zone of a given destination use reactive routing algorithm to discover routes to that node. Correspondingly, adjusting the zone radius changes the extent of proactive routing and reactive routing, and the overall routing performance is affected.

To achieve certain control requirement mentioned above, the hybrid algorithm finds the balance point automatically between proactive and reactive strategies through measuring network characteristics.
3.4 Credibility of MANET Simulation-based Studies

(such as average link lifetime and average node degree) as well as traffic characteristics (such as loss rate and delay jitter). For example, each node computes proactive overheads and approximate reactive overhead separately with an analytical overhead model. The zone radius is tuned to minimise the overall overhead by estimating the incremental difference in the overhead of the two components.

The downside of the proposed hybrid algorithm lies in its accuracy and complexity.

First of all, its performance largely depends on the accuracy of network measurement, which is still actively studied. Because of the frequent topology/traffic changes, it is very difficult to get accurate estimation of real-time network/traffic characteristics in practice.

Secondly, the complexity in zone maintenance and continuous network monitoring introduces extra processing overhead and increases the complexity in configuration (such as reconstruction intervals) and implementation. For example, the IP header of each data packet is changed in order to facilitate jitter measurement. The zone maintenance mechanism would not work well in high-mobility networks since it alters the zone radius periodically every reconstruction interval, which might fail to respond quickly to network/traffic changes.

Adaptive Distance Vector Routing Algorithm

Boppana et al [66] proposed an adaptive Distance Vector routing algorithm. Like DSDV [23], ADV exchanges route updates between the neighbouring nodes. However, only the route entries of active nodes are advertised, which reduces the size of route update messages. In addition, route updates are triggered only under certain conditions, such as route unavailability. Trigger thresholds are used to determine whether a "partial update" or a "full update" is advertised.

ADV algorithm has several shortcomings. Firstly, although adaptive to mobility conditions, the topology update advertisement and its performance are determined by "constant trigger thresholds" that need to be configured. Secondly, triggered by network events, the route update frequency might increase quickly with the node mobility, which leads to larger overheads than periodic updates. Thirdly, since only partial route information is maintained, it might take longer for a new connection to find a valid route. Due to these problems, the performance of ADV might still be under question, especially when compared with other routing protocols such as AODV and OLSR under various factors such as node mobility.

3.4 Credibility of MANET Simulation-based Studies

Simulation provides an attractive method for evaluating the performance of MANET routing protocols. For example 75.5% of MobiHoc full-paper publications during 2000-2005 used simulation to test their research [67]. However, the lack of rigor has threatened the credibility of simulation-based studies in MANET community. Surveys [67, 68] summarise the common pitfalls to avoid in conducting simulation studies. These surveys provide useful guidelines for our experimental design (see Chapter 4).

- **Model Validation & Verification.** The protocols that are being evaluated should be verified to ensure they are correctly coded and operate in accordance with the protocols' specifications. The configurable variables of the simulators must be defined.
• **Steady-state Simulation.** Prior to data collection, it is critical to determine whether the simulation has reached steady state. Data collected before reaching steady state is biased by the initial conditions of the simulation and therefore cannot be used in the analysis.

• **Data Analysis.** The correct statistical methods should be used in data analysis. A single set of data from a simulation must not be accepted as final results. It is necessary to calculate confidence intervals in order to obtain an estimated range of values.

• **Repeatability.** The simulation results should be repeatable. The source code and the raw data are expected to be publicly available.

### 3.5 Summary

This chapter started with a description of the basic signalling concepts. We then presented background information on soft-state signalling protocols and discussed several theoretical studies on soft-state performance. In Section 3.2, we surveyed several most popular routing protocols in MANETs. We discussed and critiqued recently proposed enhancements to MANET signalling in Section 3.3, including several recent studies on adaptive routing approaches.

We observe that current research efforts towards efficient signalling do not present a comprehensive understanding of the signalling performance in MANETs. The existing signalling approaches in MANET routing protocols do not provide sufficient solutions for the problems observed. The background knowledge described in this chapter sets the stage for our work. In the next chapter, we outline the methodology used in our work and describe the details of experimental configurations that serve as our research platform.
Chapter 4

Experimental Methodology

In this chapter, we describe the methodology and tools used to carry out the research reported in this dissertation. We start by describing the fundamental methodology of performance evaluation studies, followed by a discussion of various mobility models, including their major benefits and drawbacks. We discuss in details our simulation environments (including simulation tools and data analysis methods) and the experimental design (including application scenarios, mobility models used, traffic models, and performance metrics). The methods related to data analysis include data collection, data processing and result analysis. We identify several principal factors for MANET routing performance studies. Finally we conclude this chapter with a short discussion.

4.1 Methodology of Performance Evaluation

Generally, the performance evaluation can use one of the following approaches: measurement of the real system, simulation and analytical modeling [69, 70].

Measurements are possible only if something similar to the proposed system already exists. Up till now, although several MANET testbeds have been built [71, 72, 73, 74], there are still no widely accepted measurement tools or methods that prove the efficiency of MANET protocols. One possible way of measurements is through the large-scale deployment. However there might still be difficulties in carrying out the performance measurements because of cost and security/privacy issues. Considering the unavailability of real MANET systems, we use a combination of discrete event simulation and analytical modeling to estimate and approximate the real performance in this study.

Analytical models simplify the problem, help us focus on the key features we wish to investigate, and generally provide the best insight into the effects of various parameters and their interactions [70]. However, since models require many simplifications and assumptions, it is necessary to make assumptions carefully in order to maximise the applicability of the results.

Discrete Event Simulation is a software-based method that uses models of the real system. The advantages of simulation tools and simulation-based study include:

- Simulation tools offer a flexible and scalable approach to create reasonably detailed physical and link models, and to repeat and control target network conditions.

- Simulation tools offer a convenient approach in varying network parameters such as bandwidth,
latencies, error rates, workload and system scale to better understand the performance under a wider variety of conditions, than may be possible in real networks with real implementations.

- Simulation-based studies provide an easy way to test and evaluate a variety of different policies and algorithms, before implementing the most promising ones in real environments.
- Simulation results are non-deterministic and easy to repeat, which facilitates error analysis by providing large enough sample spaces of experimental results.
- Simulations can incorporate more details and require fewer assumptions than analytic evaluation and, thus, more often are closer to reality.

One potential problem of simulation-based study is that simulations may take a long time. Therefore, with simulations, it is critical to search the space of parameter values for the optimal combination and perform factorial analysis, to identify the impacts of various factors.

4.2 Mobility Models

The mobility model is designed to describe the movement pattern of mobile users on how their location and velocity change over time [75, 76]. Since mobility patterns may play a significant role in determining the performance of routing protocols, it is desirable for mobility models to simulate the movement of real-world applications in a reasonable way. Otherwise, the observations and the conclusions from the simulation studies might be misleading. Therefore, when evaluating the performance of MANET routing protocols, it is critical to choose the proper underlying mobility models.

Since MANETs have not been widely deployed and implemented, it is still difficult to obtain mobility traces. Therefore, trace-based mobility models are not widely available. Instead, various research on mobility models has been proposed to approximate real-world mobility [77].

4.2.1 Random Waypoint Model & Its Variants

The most commonly used mobility model is Random Waypoint Mobility (RWP), which is simple and easy to simulate although not very realistic. Recently, various mobility models have been proposed in order to better simulate real-world mobility scenarios than RWP. These mobility models range from the Obstacle Mobility model [78] to the City Section Mobility model [79].

All of the proposed models above suffer from non steady-state distribution at the start of a simulation [80, 81, 82]. The probability distribution of the movement of nodes typically varies with time, and ultimately converges to a steady-state distribution, known in probability literature as a stationary distribution. The convergence time varies widely, depending on the parameters of the mobility model and is not deterministic. The performance of the network varies with time, and there might be a substantial difference between start-up time after steady state has been reached. The suggested method to deal with this problem is to discard the initial set of observations hoping that steady state would have been established. Other than throwing away useful computation work, it is extremely difficult to predeterminate the length of this transient period. Yoon et al [81] recently solved the problem for RWP models by finding the steady state distribution, and initializing the mobility state to a sample drawn from the steady state
distribution. But as explained earlier, this RWP model is not realistic for the evaluation of a protocol design.

4.2.2 Random Trip Mobility Model

The Random Trip Model (RTM) \cite{83} was recently proposed as a generic mobility model that contains many particular mobility models, including the widely known Random Waypoint and Random Walk. The main advantages of RTM are listed as follows.

- **Perfect Simulations.** We are commonly interested in average-case performance of a protocol that is captured by long-run averages. If the node mobility ultimately converges to a steady-state, the long-run averages are determined by the mobility steady state \cite{84}. Simulation results collected within a short steady-state simulation period could represent long-run averages, which leaves simulation results independent of the length of simulation period.

  The major benefit of RTM is that it provides perfect simulations \cite{84}. That is, it is in steady state throughout a simulation. Therefore, this removes the need to throw away the initial set of the simulation results.

- **Real-world Map.** We are commonly interested in mobility models specified by real-world empirical mobility traces. In order to instantiate a realistic sample of a space graph, we use RTM with existing real-world city maps. United States Census Bureau makes available detailed street maps for the whole of United States, based on the bureaus TIGER (Topologically Integrated Geographic Encoding and Referencing) database \cite{85}.

- **Accommodating Common Mobility Models.** RTM implemented the RWP, Random Walk with Wrapping, Random Walk with Reflection, Restricted Random Waypoint and its special case space graph. All these models are instances of the RTM.

4.2.3 Manhattan Model

In the RWP model and other random models, the mobile nodes can move freely within the simulation field. However, in many realistic cases, especially for the applications used in urban areas, the movement of a mobile node will be bounded by obstacles, buildings, streets or freeways. The RWP and its variants fail to represent some of the mobility characteristics likely to exist in Mobile Ad Hoc networks. Therefore, in addition to the random models, we adopt another mobility model in our simulations, the Manhattan model.

The Manhattan mobility model is proposed to represent movement in an urban area. In the Manhattan model, the nodes are allowed to move along the horizontal or vertical streets on the urban map. At an intersection of a horizontal and a vertical street, the nodes can turn left, right or go straight. The probability of moving on the same street is 0.5, the probability of turning left is 0.25 and the probability of turning right is 0.25 \cite{86, 87}.

Note that the Manhattan mobility model focuses on nodes moving along horizontal or vertical streets, which may not be enough to model nodes moving in much more complex urban environments.
Therefore, it is mainly used as complimentary to the Random Trip Model, with which we can utilise real-world maps.

### 4.2.4 Reference Point Group Model

The Reference Point Group Mobility (RPGM) Model [88] is a typical group mobility model. In the RPGM model, each node in a group has two components in its movement: the *individual* component and the *group* component.

The individual component is based on the RWP model. A node randomly picks a destination within the *group scope* and moves towards that destination at a fixed speed. Once the node reaches the destination, it selects another destination randomly and moves towards it after a pause time.

The group component of mobility is shared by all nodes in the same group and is also based on the RWP model. In this case, the destination is chosen within the whole experimental area.

The drawback for such a model is that it is based on RWP. Therefore it is necessary to throw away the initial transient states from the whole simulation results.

### 4.2.5 Mobility Models Based on Social Network Theory

In addition to entity models and group models, Musolesi et al [89] propose a class of interesting models founded on social network theory. These models are based on the fact that (1) the movement of humans is strongly affected by social relationships, and (2) humans are known to associate in particular ways that can be mathematically modeled.

However, there are still some open problems associated with modeling social relationships, such as link prediction. Moreover, the models are limited to human anticipated mobility, not suitable for automated mobile devices. In addition, from the lessons of RWP, it is still not clear if the social mobility model reaches *steady-state*.

### 4.3 Simulation Environments

In the following paragraphs, we briefly describe our choice of simulation parameters, including simulation tools, application scenarios, mobility models, traffic patterns and performance metrics.

#### 4.3.1 Choice of Simulation Tools

In our simulations, we use NS, the UCB/LBNL network simulator (Version 2.8 & 2.9 ) [90]. We use the ad-hoc networking extensions provided by the Rice Monarch project [91]. The MAC/PHY modules used in this study are shown in Table 4.1.
NS is an event-driven simulator originally derived from REAL. The object-oriented software architecture of NS makes it easy to add new modules and extend existing ones using object inheritance.

Simulation scripts are written in OTcl (Object Tcl) and the core NS is written in C++. As shown in Fig 4.1, the objects in OTcl scripts and C++ codes are bound using a special bind procedure. Any modification to the objects initiated from one language is visible in the other. This allows access to the objects from either language and makes it easy to move functionality between these two programming realms.

The choice of languages involves a trade-off between performance and ease of use. OTcl makes it easier to rapidly prototype new algorithms, while C++ is more suitable for large simulations that require high performance. Therefore, functionality that requires per-packet processing in NS should be implemented in C++, while the experimental code fragments that are not frequently used can be implemented in OTcl. Since this study is on routing protocols, we mainly use C++ in implementation.

Other popular simulators, such as OPNET Modeler and GloMoSim, also provide advanced simulation environments to test and debug any kind of networking protocols, including wireless applications. In this study, we choose NS instead of other simulators for the following reasons.

- Compared with OPNET, NS is fully open-source. With the source code we could verify the correctness of the implementations and debug the source of possible errors. Furthermore, open-source simulators facilitate measuring internal metrics such as route durations.

- Among these three simulators, NS provides the most comprehensive implementations, especially for MANET routing protocols. Up till now there have been no OLSR implementations on GloMoSim, while at least three different OLSR implementations exist on NS.

- NS has the largest user community; it is therefore much easier to get help and support from online FAQs/tutorials and other online communities such as mail lists.

4.3.2 Choice of Application Scenarios
We use two types of MANET scenarios in our simulations: synthetic scenarios and concrete scenarios.

In synthetic scenarios, parameters such as mobility, traffic load and refresh intervals are varied over an arbitrary range of values. Synthetic MANET scenarios provide a range of comprehensive simulation environments, which enable us to carry out performance evaluation by subjecting the protocols under
4.3. Simulation Environments

consideration to more stringent network conditions such as higher mobility and higher density, as well as a variety of routing characteristics (e.g. radio ranges and route length).

We also define concrete simulation environments that reflect specific MANET applications, namely urban scenario, rescue scenario and conference scenario. These concrete scenarios represent diverse aspects of real-world MANET applications and services, e.g. urban scenario for restricted mobility pattern, rescue scenario for mission-critical applications and conference scenario for a group based mobility pattern. Results obtained from concrete scenarios are considered more realistic in terms of real-world deployment. The following paragraphs give a detailed description on these concrete scenarios.

- **Urban Scenario** represents MANET applications and services deployed in restricted areas with obstacles, such as metropolitan security operations in urban areas. In such scenario, nodes can only move along certain specific routes with possible barriers, instead of random movements in an open area. This scenario reflects that in a large network, it is less probable that, for each movement, a node selects a random destination within a very large geographic area.

  A typical urban scenario consists of slow-speed moving objects such as policemen walking on foot (0-2m/s) and relatively high-speed moving objects such as police cars (0-20m/s). The pause time is 0-5s. The mobility models adopted include Manhattan model and restricted random waypoint model (taken from the Random Trip model). The nodes are only allowed to move along specific routes, e.g. the horizontal or vertical streets in the Manhattan Model, or routes from topologies based on towns and highways of real-world maps in Restricted RWP model.

- **Rescue Scenario** describes mission-critical MANET applications, such as disaster and emergency rescue operations. Under such scenarios, the nodes' movement is usually random and unpredictable; the velocity of mobility objects is diverse, from slow movements such as walking speed to high-speed movements such as airborne movements; natural surrounding circumstances are often bad, such as thunderstorms in a martial rescue operation, which may impact the wireless signals and communication quality and cause packet loss. Moreover, MANET applications and services are required to provide high delivery reliability and timelines' guarantees in the presence of mobility and permanent or temporary resource outages.

  A typical rescue scenario consists of slow moving objects such as rescue staff that searches the field randomly on foot (0-2 m/s), rescue vehicles moving at medium speed (5-20m/s) and airborne objects such as helicopters which move at relatively high speed (10-60 m/s). Each team covers well-defined areas within the field with sufficient overlap to ensure the information can be relayed among the different teams. The mobility model is the Random Trip Model.

- **Conference Scenario** represents MANET applications and services within offices and communities. Under this scenario, the node movement is usually group-based; the movement speed is relatively low (0-5 m/s) with a long pause time; the movement pattern is not random but restricted within groups.

  A typical Conference scenario consists of one speaker node and three groups of audiences, i.e.,
4.3. Simulation Environments

audience 1, audience 2 and the audience 3. All of the audience groups consist of a number of members (20 for example) moving with speeds between 1-3 m/s. The movement of the audience groups is modeled using the Reference Point Group Mobility model and node movement is restricted to a limited area within the field. The 3rd audience group consists of nodes that are capable of moving over the entire topology. The speeds for these nodes were randomly chosen between 1-5 m/s with pause times between 0-5 seconds. The Random Trip Model is used as the mobility model for the 3rd audience group.

4.3.3 Choice of Mobility Models

In order to obtain realistic simulation results, it is necessary to examine the routing performance under multiple mobility models. Generally, various mobility models could be classified into two types, e.g. constrained-topology models and statistical models.

- **Constrained-topology models** simulate real-world scenarios but still have some randomness to provide variability. In our study, we adopt a wide range of constrained topology based models, including the Freeway model, the Reference Point Group Mobility model (RPGM), the Manhattan model and the Restricted Random Waypoint model (RRWP, part of Random Trip model). The last two models use real-world maps to generate mobility patterns. For example, RRWP uses existing American city maps available from TIGER (Topologically Integrated Geographic Encoding and Referencing) database of the United States Census Bureau.

- **Statistical models** are featured with full randomness. Nodes can move to any destination with randomly chosen velocities and directions. Although idealistic, this type of model provides common test beds for performance comparison of different routing protocols and also for emerging services in which the nodes' real mobility pattern is not known or foreseeable. In our study, we use the Random Trip model, which provides steady-state simulations (e.g. "Perfect Simulations") and removes the need to discard the warm-up period of results.

Note that, some of constrained topology based models share the same problem with RWP. In order to guarantee the steady-state property of the simulations, the minimum speed of these models is set to be non-zero. The results from the initial simulation period are discarded [81].

4.3.4 Design of Traffic Patterns

CBR (Constant Bit Rate) traffic is unreliable, unidirectional, and predictable with known packet size and packet interval and stream duration. In our study, we choose CBR traffic because measuring CBR traffic provides the most straightforward and accurate method to evaluate the performance of routing protocols.

This study is to investigate the packet delivery performance of MANET routing protocols. In order to evaluate the routing performance, we measure metrics such as throughput and traffic overhead. CBR traffic meets such requirements. Transport-layer traffic like TCP is not suitable for our study because its inner mechanisms on rate/congestion control and reliable transmission would impact the behaviour of data traffic and increase the complexity of data analysis.
Simple CBR traffic patterns that do not support simultaneous connections per source might be unrealistic. Unfortunately, traffic-generation parameters depend on specific applications and have a wide range of possibilities, such as packet size, variable versus constant bit-rate generation.

A random distributed CBR traffic model is used in this study, which allows every node in the network to be a potential traffic source and destination. The CBR packet size ranges from 50 bytes to 1000 bytes. The CBR rate is from 10 kbps to 100 kbps. Without a universal or driving application, we properly document the simulated settings so that the results can be duplicated independently.

4.3.5 Performance Metrics

For each performance study, a set of performance metrics must be chosen. Generally speaking, a system's performance is usually measured in time (or responsiveness, i.e. the time taken to perform the service), rate (or productivity, i.e. the rate at which the service is performed) and resource (or utilisation, i.e. the resources consumed while performing the service) [69]. In particular, RFC2501 [92] defines a list of quantitative metrics that can be used to assess the performance of MANET routing protocols. In this section, we present a discussion on how we select performance metrics for our performance study.

- **Packet Delivery Ratio (PDR)** is considered as the most straightforward metric for MANET routing protocols. PDR is the percentage of data packets that are successfully delivered, calculated as the number of data packets received by the destination nodes divided by the number of data packets sent by the source nodes.

In each simulation of this study, the number and the size of data packets sent by the source nodes is kept constant. This allows end-to-end throughput to be used instead, which plays the same role as PDR.

In communication networks, end-to-end throughput is the amount of data per time unit (i.e. data rate) that is successfully delivered through a network, from one node to another via communication links. In this study, it is computed as the amount of data transferred (in bytes) divided by the simulated data transfer time (the time interval from sending the first data packet to receiving the last data packet). The throughput of a routing protocol is the average data rate that is successfully delivered through a network that runs the routing protocol.

Since routing is to discover/select paths in a network along which to send data packets, throughput and PDR measure the productivity or effectiveness of a routing protocol. That is, they measure the amount of output created (i.e. data packets received) per unit input used (i.e. per data packet sent). This study is to propose effective signalling algorithms to improve packet delivery performance of MANET routing protocols. Therefore, the performance of our proposals can be best evaluated by comparing the routing throughput (or PDR) of the proposed algorithms with that of the original protocols.

- **Control Traffic Overhead** measures the amount of resources consumed (i.e. the cost) while performing the routing operations. The resources usually include battery life and bandwidth. In this study, we only consider bandwidth consumption, i.e. control traffic overhead.
In order to gauge the routing protocol overhead, we measure both the number of routing messages and the number of bytes in the routing packets transmitted. Normalised routing overhead (NRO) is computed in order to show the routing efficiency. \( NRO \) is defined as the ratio of the number of control packets propagated by every node in the network, \( P_c \), to the number of data packets delivered through the network, \( P_d \).

\[ NRO = \frac{P_c}{P_d} \]  

(4.1)

Because of the broadcast nature of control overhead transmission, the overhead of a MANET routing protocol is calculated by the volume of routing control packets being received by each node in a mobile ad hoc network, measured by the number of routing packets and the amount of the routing packets by bytes. The control overhead gives a baseline evaluation of the system cost. Basically the control traffic overhead generated by the MANET routing agent is to set-up and to maintain the route state. From this point of view, two factors contribute to the increase of the control traffic overhead: the size of the control packets (i.e. the number of nodes) and the number of hops over which the control packets are forwarded (i.e. the distance of transmission).

- **End-to-End Delay** refers to the time taken for a packet to be transmitted across a network from source to destination. It is determined by the size of the data packet and the number of hops. Delivering data packets along a shorter path usually leads to smaller end-to-end delay. Therefore, end-to-end delay is another important performance metric of routing protocols.

In this study, the proposed signalling algorithm has no direct impacts on the end-to-end delay. The following example gives a further explanation.

As shown in Fig 4.2, in case (1), node \( F \) moves towards the other nodes and establishes a shorter path \( A \rightarrow B \rightarrow F \rightarrow E \) between node \( A \) and node \( E \). The proposed signalling algorithms help detect the link establishments, and therefore packets sent from \( A \) to \( E \) follow the new path, which reduces the delay in packet delivery.

However, as shown in case (2) of Fig 4.2, node \( F \) moves away from the nodes, which leads to the breakage of path \( A \rightarrow B \rightarrow F \rightarrow E \). The signalling algorithms discover the alternative path \( A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \) quickly and reduce packet drops. On the other hand, the average delay in packet delivery increases since the saved data packets are delivered along a longer path.

In addition to these performance metrics, we also measure several internal variables, including *link duration* and *route duration*. *Link duration* is the maximum period during which the two nodes are within the transmission range and there exist symmetric links between the two nodes. *Route duration* of proactive routing protocols is the maximum period during which (a) each of the links of the route between the nodes is connected and symmetric; (b) the route keeps the shortest distance between the two nodes.

In routing performance research, the workload is defined as the data traffic delivered in the network.
4.3. Simulation Environments

The execution of routing functions for each node can be represented by a utility function, which quantifies the level of satisfaction a node gets from using the network resources. The utility function takes into account the energy that a node spends for the purpose of data packet forwarding $E_D$ (including its own communications and relaying data packets on behalf of other nodes) and the energy that the node has to use when participating in the routing protocol $E_R$ (including neighbour detection $E_{nb}$ and topology advertisement $E_{topo}$). The utility function used to study the signalling mechanisms chosen by a node is the following:

$$U(p, s_{nb}, s_{topo}) = E_D - E_R$$

$$= E_D - (E_{nb} + E_{topo})$$

$$= p\lambda E_d - s_{nb}E_{nb} - s_{topo}E_{topo}$$  (4.2)

where $p$ is the probability that a data packet is delivered to its correct destination, $s_{nb}$ is the rate of control packets in neighbour detection and $s_{topo}$ is the rate of control packets in topology advertisement.

The other factors that appear in the utility function are respectively:

- $E_D$, energy spent for data packet sending and forwarding.
- $E_R = E_{nb} + E_{topo}$, energy spent for participating in the routing protocol.
- $E_d$, energy required to send or forward a data packet.
- $\lambda$, data flow rate.
- $E_{nb}$, energy required to send a neighbour detection message.
- $E_{topo}$, energy required to send or forward a topology advertisement message.

### 4.3.6 Assumptions

In order to simplify the simulation scenarios, we have to propose multiple assumptions in our simulation-based studies. Several typical assumptions are listed as follows.

- Transmission range is generally represented as a circle's radius.
- Node distribution is usually modeled as uniform or random. In reality, roads, trees, water, and other obstacles affect node distribution.
- Interference models are typically based on SNRs (Signal-to-Noise Ratios) or BERs (Bit Error Ratios), which neglects interference based on increasing traffic or unpredictable background noise.
Nodes may move randomly. However, random node mobility is rare in reality. Individuals rarely travel in random directions or pause for random times. Instead, they usually follow some patterns.

- The simulation area is usually a square or rectangular area with boundaries. And nodes never move out of the defined areas.

Our study addresses some of the assumptions listed above. For example, node distribution and node mobility in our simulations are not totally randomised. We use real-world maps to construct node mobility patterns [83]. However, it is not possible to eliminate all of the assumptions. Therefore, we keep in mind the effects of the assumptions on the performance. The simulation based performance analysis would not be carried out before the sources of the performance differences (or improvements) are not fully understood.

The settings of the protocols share similar situations with assumptions. It is not possible to give all possible values to the protocol parameters. Instead, the impacts of the settings in our simulations are clearly understood and the choices of the settings have been well documented.

### 4.4 Data Collection, Processing and Analysis

#### 4.4.1 Data Collection

There are a number of ways of collecting results from a simulation. In this study, we collect trace data generated by the simulations for the subsequent statistical data analysis.

Generally, trace data is either displayed directly during execution of the simulation, or more commonly stored in a file to be post-processed and analysed. There are two primary types of monitoring capabilities supported by the NS simulator.

The first method is called traces, which records each individual packet as it arrives, departs, or is dropped at a link or a queue. Trace objects are configured into a simulation as nodes in the network topology.

The other method is called monitors, which records counts of various quantities such as packet and byte arrivals, departures, etc. Monitors are supported by a separate set of objects that are created and inserted into the network topology around queues. They provide an easy way to calculate arrival statistics.

Compared with monitors, traces are more generic and extensible. It is easier to use traces to implement customised data collection methods.

In this study, we use traces to record network events and perform post-processing. NS supports a set of extensible trace mechanisms. With the default implementation of CMU wireless extension, the trace file records standard network events, including packet sending, receiving and dropping events. In addition, NS provides interfaces to facilitate the developer recording user-defined network events into the trace file. Correspondingly, we use the following two types of tracing methods.

- Standard trace formats generated by existing NS implementations
The details of the standard NS trace format can be found in Appendix A. Here is an example trace item from our simulation traces:

```
s -t 0.000169789 -Hs 0 -Hd -1 -Ni 0 -Nx 0.00 -Ny -1.69 -Nz 0.00 -Ne -1.000000 -N1RTR -Nw — -Ma 0 -Md 0 -Ms 0 -Mt 0 -Is 0.255 -Id -1.255 -It OLSR -D 48 -If 0 -Ii 0 -Iv 32 -P olsr -Pn 1 -Ps 0 [-P HELLO -Po 0 -Ph 0 -Pms 0]
```

Figure 4.3: Standard Trace Example

From the above trace item, we could obtain the following information:

A HELLO message of OLSR agent is broadcasted by node 0 at time 0.000169789; the packet length is 48 bytes; it is a routing control packet, instead of data packet.

With these packet-sending/receiving/dropping events, we could calculate throughput and control overhead of the routing protocols.

- Customised trace format

Based on existing NS implementation, we add self-defined loggers into each routing agent to record any interesting network events. Here is one customised trace item from our simulations:

```
c -t 4.235174208 -Ni 2 -o LOCAL_ADD_NB -Nb 15
```

Figure 4.4: Customised Trace Example

From this we know that, node 2 detects a new neighbour node 15 at time 4.235174208.

Note that, when making the extensions, we avoid conflicts between our extensions and the standard trace formats.

### 4.4.2 Data Processing

One problem with trace based data analysis is the large volume of trace files. The size of a typical trace file, generated by one single high-density network simulation run (for example, a network with 50 nodes), is around 200 Mbytes. Due to the requirements of statistical data analysis, we usually need to generate 30-80 such trace files in order to obtain one data point. Therefore, considering the large cost in storage, the trace files are processed immediately after each simulation.

In order to extract performance information from the trace files, we built a suite of extensible data processing/analysis tools as a toolkit. Given each NS trace file as input, the trace analysis toolkit resolves the routing events by matching the tags in each trace item. In the following paragraphs, we introduce briefly the basic data processing mechanisms of this study.

The main functional trace-analyser scripts are written in python, an interpreted, interactive, object-oriented, extensible programming language with rich library supports. The other scripts are written in Unix Shell (bash) scripts. These scripts execute the simulation tools (e.g. ns2) to generate trace files and call the trace-analysing functions in the python scripts to analyse the trace data.

The trace processing is described as follows.
4.4 Data Collection, Processing and Analysis

1. The bash shell script executes a simulation; trace file is generated.

2. The bash script calls trace-analysing functions to extract data from NS trace file; the trace file is the input of the functions.

3. The python scripts analyse the trace file by matching the tags; the characteristics of each data flow are stored in an array; the average throughput and the average overhead of this simulation are calculated and stored into a temporary file.

4. Go to (1), until the experiments required have been finished.

5. The python scripts read the performance results of each simulation from the temporary file and carry out error analysis based on standard statistical methods, e.g. calculating the mean value and the standard errors.

4.4.3 Data Analysis

After obtaining the simulation results, we carry out further statistical analysis on performance metrics (such as throughput and overhead) and internal variables (such as link duration and route duration).

Statistical results based on multiple simulation runs improve the accuracy of the results. Each simulation experiment comprised a number of independent trials. In addition, different mobility models were used to further explore the system characteristics. We calculate the mean value of simulation results and the confidence intervals at 95% confidence level exploiting the Central Limit Theorem (CLT). As the number of trials increases, variance of data samples decrease, which allows better mean value estimation and provides a more precise range or deviation.

In addition to providing an expected value range, confidence intervals help determine if two data sets are statistically equivalent. If the mean value of simulation results falls within the confidence interval of the measurement results from the real experiment and vice versa, the simulation results are statistically equivalent to the real-world experiments.

In a statistical data analysis, it is essential to determine the number of required independent runs. The Central Limit Theorem can determine the number of simulation runs required. Each simulation of this study is started by performing a small number of independent runs. Then we calculate the mean and standard deviation of the results that are used to determine the number of simulation runs necessary. An initial study found that a sample size of 30 should be sufficient. However, in order to provide better estimation, the number of simulation runs was set to be 100.

In order to study the approximate distribution of the internal variables (i.e. link duration and route duration), we use the relative frequency approach from probability theory to estimate the PDFs (Probability Density Functions) and CDFs (Cumulative Density Functions) of the variables across the different mobility models. Once the PDFs are determined, we calculate the average value of the variables based on the probability. In order to approximate the distribution, we use standard curving fitting technique to analyse the PDFs of the variables. The results of the curve fitting, including the Root Mean Squared Error (RMSE), are listed in tables to facilitate further analysis. After fitting data with one or more models, it is necessary to evaluate the goodness of fit.
Data analysis is part of the extensible data processing/analysis toolkit. With the toolkit, the statistical results are generated immediately after each simulation, including the mean and the errors. No trace files are stored.

4.5 Factorial Analysis

The factors in this performance study include mobility, traffic load and protocols.

- **Mobility.** The factor of mobility includes the types of mobility models adopted and the parameters of these mobility models, including the (maximum/mean) node velocity and the node density (i.e. node number and the size of the area).

- **Traffic Load.** The factor of traffic load refers to the parameters of the CBR traffic, including the rate of CBR and the size of the packets being sent.

- **Protocol Parameters.** The factor of the protocols includes the refresh intervals of routing protocols (i.e. HELLO interval and TC interval for OLSR, and update interval for DSDV) and the variables of low-level protocols (i.e. physical layer and MAC layer), including transmission ranges.

The goal of factorial analysis is to understand the impact of each factor on the performance metrics. In following paragraphs, we focus on the factorial design (including the range of the factor value) and the data variance under these factors.

4.5.1 Factors

- **Mobility Model.** In order to obtain realistic simulation results, it is necessary to examine the routing performance under multiple mobility models. The mobility models studied in this research represent various existing models, either constrained topology based models or statistical models. Specifically, the constrained topology based models of this study include the Freeway model, the Reference Point Group Mobility model (RPGM), the Manhattan model and the Restricted Random Waypoint (RRWP) model. The statistical model of this study is the Random Trip Model.

One important criteria in selecting mobility models is the steady-state simulation. Since we are commonly interested in measuring the performance of a protocol that is captured by long-run averages. If the node mobility ultimately converges to a steady state, the long-run averages are determined by the mobility steady state.

It is expected to observe the same behaviours of the proposed systems under various mobility models. Specifically, the value of the performance metrics may be slightly different under different models; however, the curves of the performance metrics should follow the same trends.

- **Node Velocity.** The value of velocity in this study varies from low-level mobility (0-1 m/s average walking speed), moderate level mobility (5-10m/s average vehicle speed) and high-mobility (20-30m/s airborne speed such as helicopters).
With the increase of the node velocity, the throughput is expected to drop because of the increase of link breakage events. In particular, the effect of velocity in low-density, small-transmission-range networks with small refresh rate is expected to be more significant compared with that in high-density, long-transmission-range networks with large refresh rate, since the links are more easily broken and the failure detection latency is greater.

- **Traffic Load.** In order to simulate IP traffic in the real world, the size of CBR packets ranges from small packets (50 bytes) to large packets (1000 bytes). The CBR traffic of each data flow ranges from 10k bytes/s to 100k bytes/s. In multi-hop wireless networks, as long as the buffer size at each node is reasonably large (say, larger than 10 packets), buffer overflow-induced packet loss is rare and packet drop due to link-layer contention dominate [93]. Further, the gradual link-drop behavior is shown as network load increases. In addition, when the bit-error-rate is relatively high (e.g. $10^{-5}$ or higher), packets of larger size have a greater possibility to be retransmitted, while packet of smaller size is not efficient because of the fixed overhead required per packet.

With the same amount of load, the number of data packets sent in small-packet CBR traffic is much larger than that in large-packet CBR traffic. Therefore, the increase of load, either data traffic load or control traffic load, would lead to more significant link-layer contention and therefore increased packet loss rate. Correspondingly, the effect of load has a more significant impact on small-packet CBR traffic.

When the network load is relatively low and the refresh rate of routing protocols is relatively low, the link-layer contention is not frequent; therefore, the effect of packet size is not significant. When the network load is relatively low and the refresh rate of routing protocols is relatively high, the link-layer contention causes larger link drop rate for CBR traffic with small packet size (since the packet number is greater) than for CBR traffic with large packet size.

In our traffic design we vary the size of CBR packets and the number of CBR connections from one single node. Such traffic patterns encompass the traffic patterns in a large set of potential applications and represent a large proportion of real-world traffic.

- **Transmission Range.** Without losing generality, we assume the transmission range (i.e. radio range for all nodes) in a mobile ad hoc network is identical or similar. The value of the transmission range of each node ranges from 50 m to 400 m in this study. The larger the transmission range is, the less likely the link gets broken due to mobility. Therefore, the increase of transmission range leads to better performance.

- **Refresh Intervals.** The value of refresh intervals ranges from 1s to 45s. Intuitively, the traffic overhead increases linearly with the increase of refresh rate. In terms of end-to-end throughput,
4.5. Factorial Analysis

- When the network is not overloaded (i.e., data rate does not exceed available bandwidth [94]), the increase of refresh rate reduces the failure detection latency and re-routing latency, and therefore improves the data packet delivery performance. However, increasing refresh rates introduces extra control overhead, which intensifies channel contention and might lead to performance degradation.

- In an overloaded network (i.e., data rate exceeds available bandwidth [94]), the increase of refresh rate may slightly improve the performance when the refresh rate is relatively low. Then it would downgrade the performance sharply because of the introduced control packets.

4.5.2 Factor Selection

Various factors may have inter-dependency or interaction between each other. That is, the effects of factor A may depend on the settings of factor B. Because of the (possible) interactions between the factors, it is necessary to use full-factorial experimental design. However, due to the large number of the experiments, we adopt the following three ways [69].

- **Reduce the number of levels for each factor.** In some cases, only two levels are required for each factor to determine the relative importance. For example, the node density can be low density and high density. The number of the levels is rather flexible. For example, the node velocity can be three levels, namely low mobility, moderate mobility and high mobility.

- **Reduce the number of the factors.** The non-critical or non-relevant factors are removed from the analysis in order to simplify the model.

  Example 1: according to a recent study on the impact of multi-hop wireless channel on throughput [93], as long as the buffer size at each node is reasonably large (say, larger than 10 packets), buffer overflow-induced packet loss is rare and packet drop due to link-layer contention dominate; further, a gradual link drop behavior is shown as network load increases. Therefore, the size of the queue buffer is not considered as a factor in this study.

  Example 2: we assume the parameters in a network, such as low-level MAC/physical layer protocols (including node transmission range), are the same for each node; even if the parameters in a real system are different, we could still assume in a relatively small area, the above parameters are similar or the same.

  Example 3: we assume the battery life of each node is infinite, so that there is no need to consider the energy issue and the problem of node power-off caused of energy exhaustion; instead, we only consider the number of packet dispatch as the factor related to energy issue.

- **Use fractional factorial design where applicable.** We can use only a fraction of the full factorial design. This is done by careful analysis on the interactions among the factors. There are no interactions between some factors, while some interactions between factors are known to be negligible.
For example, the control overhead largely depends on the refresh rate of the routing protocols. The effect of various mobility models on the overhead could be treated as the same, since the variance of the loss rate of control overhead under various mobility models is negligible (assuming other factors such as node density and movement speed are the same). Therefore, there is no need to evaluate the overhead under various mobility models.

4.6 Summary

In this section, we describe briefly the design of the proposed experimental studies. In order to approximate the performance measurement of real-world mobile ad hoc networks, we propose realistic application scenarios and mobility patterns, and appropriate traffic patterns and performance metrics. These efforts in designing the experimental environments have made fundamental contributions to the following work on performance evaluation.
Chapter 5

Understanding the Impact of Soft Intervals on the Performance of OLSR and DSDV

The simplicity and robustness of the soft-state approach makes it an attractive choice in the state maintenance of a variety of protocols and applications. This is especially true for MANETs where frequent link failures and topology changes due to node mobility makes it difficult to maintain state consistency. However, despite its wide deployment, there is still no comprehensive understanding or well-accepted model of the performance (resilience, robustness etc.) of soft state protocols. The performance of soft-state signalling is not well understood in the context of MANETs.

In this chapter we present a quantitative analysis of the impact of a periodic refresh update strategy on the routing performance of proactive MANET routing protocols. Using an analytic resilience model based on probability theory, we infer that the signalling performance depends largely on environmental factors like node mobility. Simulation results support our findings and show that the temporal state updates have a significant impact on the throughput, while the topological state updates do not. This establishes a clear understanding of how soft-state signalling impacts performance and gives guidance in designing new applications and network protocols.

5.1 Overview

In the soft-state approach, the state information stored in a node (i.e. state holder) is removed unless a periodic refresh message from the state installer is received, confirming the state validity. Protocols using soft-state signalling therefore do not require any explicit state removal mechanism as part of their state maintenance. If the state installer is unreachable due to either connection failure or node crash, no actions are required to remove "orphaned state" [33]. This significantly reduces the interaction complexity between different distributed entities. Moreover, since loss of signalling messages can be covered by the subsequent refresh messages, the soft-state approach simply uses best-effort traffic in signalling and does not require reliable message delivery mechanisms.

Features like simplicity and robustness make soft-state signalling a favorable choice for a variety of network protocols and applications, especially for dynamic networks such as MANETs. Many MANET routing protocols exchange periodic HELLO messages between neighbouring nodes in order to detect...
5.1. Overview

link changes (i.e. link breakage and establishment) and to maintain neighbour status. This strategy is not limited to proactive protocols like OLSR (Optimised Link State Routing) [16], DSDV (Destination Sequenced Distance-Vector Routing) [23] and TBRPF (Topology Broadcast based on Reverse-Path Forwarding) [24]. It is also used in reactive protocols like AODV (Ad hoc On-Demand Distance Vector Routing) [25] and TORA (Temporally-Ordered Routing Algorithm) [26] and even in hybrid protocols like ZRP (Zone Routing Protocol) [27]. Moreover, proactive routing protocols use periodic network-wide update messages as part of their topology advertisement operations.

Existing studies on soft-state performance [33, 30, 31] are mostly based on generic mathematical models applied to fixed networks. Despite their fundamental difference, there have been no studies of the performance of soft state protocols in the context of mobile ad hoc networks. Considering the significant difference between fixed networks and MANETs, a thorough investigation of soft-state signalling performance is required. The properties of soft state signalling in such volatile networks need to be fully understood in the presence of frequent network changes, especially its impact on systems and the circumstances in which it might best be employed. For example,

- **Timer intervals** of soft-state signalling protocols are usually configured manually based on recommendations of original protocol designers or experiences. There have so far been no comprehensive investigations or theoretical studies on how to determine those intervals. The intervals are kept unchanged no matter what network conditions are. Such fixed timer intervals might work in less volatile networks (such as fixed networks). However, it is not clear if the default value of timer intervals performs well with failures in dynamic & heterogeneous networks like MANETs. Moreover, it is not obvious how to estimate the refresh intervals, under given service requirements, that can achieve the best balance between performance and control traffic overhead.

- It is commonly believed that a smaller refresh interval in soft state mechanisms could speed up the adaptation process to network changes at the expense of increased overhead. However, there is no comprehensive study of how much it could improve the consistency. Such information is essential for proactive MANET routing protocols, since network dynamics require effective signalling to maximise throughputs, while resource constraints require minimised control overhead to reduce channel contention and battery consumption. The soft-state performance thus needs to be quantified in order to enhance system efficiency and scalability.

- Soft-state signalling enables MANET nodes to maintain a consistent view of network topology by propagating periodic topology updates. However, link or topology changes might be too dynamic to be captured by periodic updates. For example, a link breakage that occurs right after a topology update propagation has to wait for a period of one topology update interval to be advertised. In the presence of frequent changes, the performance of soft-state signalling needs to be re-evaluated.

In this chapter, we present a quantitative analysis on the impact of periodic refresh updates for proactive MANET routing protocols and assess its impact on routing performance. In particular, OLSR
5.2. Soft State Signalling in Proactive MANET Routing Protocols

5.2. Soft State Signalling in Proactive MANET Routing Protocols

In this section, we introduce briefly background information on soft-state signalling in proactive MANET routing protocols, including OLSR and DSDV.

5.2.1 Optimised Link State Routing

The Optimised Link State Routing Protocol (OLSR) [16] is a typical LS protocol. It uses proactive routing to maintain the routing information about each node in the OLSR network. OLSR nodes detect changes in the connectivity to their neighbours by exchanging periodic HELLO messages. Topology Control (TC) messages are propagated among all the nodes of the network to discover available routes in the presence of mobility and resource constraints.

Signalling Messages

Like the QoS signalling messages in RSVP, signalling messages in OLSR can be categorised into two types: trigger messages and refresh messages [95, 96].

Trigger messages are generated due to state changes, including initiating new state (i.e. new neighbours and links) and removing obsolete state. For example, the first HELLO message received from a new neighbour is a trigger message that initiates a neighbour state in the neighbour list. Neighbour sensing [16] in OLSR is a trigger process that discovers new neighbours and links. OLSR nodes remove obsolete neighbour state either by state time-out, or by link layer notification. In addition to initiating new link state in the topology repositories of each node, the topology advertisement process in OLSR removes obsolete topology state implicitly by assigning sequence numbers to topology advertisements.
5.2. Soft State Signalling in Proactive MANET Routing Protocols

Refresh messages contain replicated state information used to update state for robustness.

An in-depth look into the use of soft-state signalling in OLSR shows that, in addition to traditional functionalities of state maintenance (i.e. state recovery and state removal), soft state plays a more important role in advertising state changes. Receiving a (periodic) HELLO message from an unknown node indicates the arrival of a new neighbour, which in turn facilitates link establishment. Instead of propagating an instant topology update, the nodes encapsulate multiple topology changes (since the last update) into a single topology update message. This reduces significantly the control overhead in the presence of frequent topology changes.

State Repositories

In order to maintain coherent topology information of the entire network in the presence of mobility and failure, an OLSR daemon needs to maintain two types of state repositories in its internal tables, namely a local state repository and a topological state repository.

The local state repository records and keeps the localised information updated, including link tuples, neighbour tuples and MPR tuples. More details about these state repositories can be found in Chapter 3.

The topological state repository (or topology set as defined in OLSR [16]) maintains the topology information of the whole network. This information is acquired from Topology Control (TC) messages and is used for the calculations of routing tables.

State Update Mechanisms

Each node in a MANET regularly transmits state updates in order to enable other nodes to maintain correct routing information. As the state information may time-out unless it is refreshed, it is called soft state. In particular, nodes using proactive routing protocols use two types of soft state updates: temporal state updates for localised neighbour detection and topological state updates for consistent connectivity information for each node. This study compares the impact these two state updates have on proactive routing performance in OLSR.

**Temporal State Update.** OLSR nodes update the local state repositories by exchanging HELLO messages. The rate of these temporal state updates is the sending rate of HELLO messages. OLSR nodes generate periodic HELLO messages based on link state information in their link set (see Chapter 3). When receiving a HELLO message, OLSR nodes update the link set first. The other state repositories are then synchronised or re-calculated by examining the link tuples in their link set.

In addition to time-out timers, OLSR nodes can use link layer notification to remove broken links from their link repository (see Chapter 3). When a data packet fails to get disseminated by the MAC layer, the node infers that the link is broken. A recent study on link sensing by Voorhaen and Blondia shows that, cross-layer supported link sensing outperforms the pure soft-state based link sensing mechanisms [20].

**Topological State Update.** OLSR nodes update their topological state repository via TC messages. The rate of the topological state updates is the frequency of TC messages.

In OLSR, only MPR nodes generate topology control messages based on the MPR selector set.
Also, only MPR nodes are responsible for forwarding control traffic, intended for dissemination into the entire network. MPRs provide an efficient mechanism for flooding the control traffic by reducing the number of message transmissions.

Similarly, in addition to timeout timers, OLSR removes obsolete topology state entries implicitly by assigning a sequence number called ANSN (Advertised Neighbour Sequence Number) to advertised neighbour sets, to keep track of the most recent topology changes. Every time a node detects a change in its advertised neighbour set, it increments this sequence number in a wraparound manner. When a node receives a TC message, it simply removes the topology information that originated from this node with a smaller ANSN value.

It is clear that the maintenance of the internal state information held at the nodes is directly related to the exchange of HELLO and TC messages and therefore anything that is affected when these messages are generated, such as refresh timer intervals, is likely to impact on protocol performance.

### 5.2.2 Destination-Sequenced Distance Vector Routing

Compared with OLSR, DSDV does not have separate state repositories. Each DSDV node maintains the neighbour state in its routing table as one-hop destination nodes. When a link to a next hop is broken, any route through that next hop is immediately assigned a metric of \( \infty \) so that it should not be selected for data delivery. Instead of a topology repository, each DSDV node maintains routing state in its routing table containing the distance from itself to all other nodes in the network. Each node broadcasts periodically its routing table to each of its neighbours and uses similar routing tables from neighbouring nodes to update its table.

Correspondingly, DSDV does not carry out separate operations for neighbour detection and topology advertisement. New neighbours can be detected when their first routing table updates are received. Neighbour state is timed out and removed if no routing table updates have been received for a period from an existing neighbour. Topology changes are advertised through exchanging routing tables between neighbouring nodes. DSDV nodes do not forward the routing tables received to their neighbours. So whenever network topology changes, it takes more time for the network to converge, compared to OLSR.

### 5.3 A Soft-State Model Based on Probability Theory

In this section, we present a soft state model based on probability theory. Through a model-based theoretical study, we aim to give a quantitative relationship between system performance and factors like refresh interval and failure rate.

In the following theoretical study, we analyse the state inconsistency time \( L \), i.e. the period from the change occurrence (i.e. when an inconsistency occurs) to the time the nodes in the network update the state repositories (i.e. achieving state consistency again). We assume that the arrival of a change event (either link changes or route changes) is an independent, identically distributed Poisson process with arrival rate \( \lambda \). The assumption is reasonable, if the node degree is small and the nodes are moving randomly so that the process of route change is totally random.
Table 5.1: Notation in Soft State Model

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>Refresh interval</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Change arrival rate</td>
</tr>
<tr>
<td>$L$</td>
<td>State inconsistency time</td>
</tr>
<tr>
<td>$\phi$</td>
<td>State inconsistency ratio</td>
</tr>
<tr>
<td>$\psi$</td>
<td>Derivative of $\phi$ with respect to $r$</td>
</tr>
<tr>
<td>$p$</td>
<td>Channel loss probability</td>
</tr>
<tr>
<td>$E[Y]$</td>
<td>Expected value of a random variable $Y$</td>
</tr>
</tbody>
</table>

5.3.1 Soft State Model without Channel Loss

Consider an arbitrary period, starting at $t_0$. Let $X$ be the time of first link change occurrence after $t_0$. Let $\gamma = X - t_0$ be the waiting time until the first occurrence after $t_0$. Let $r$ be the refresh interval.

We define $L$ as,

$$L = t_0 + r - X$$

And we have,

$$P(\gamma > t) = e^{-\lambda t}$$

In probability theory, if the probability distribution of $Y$ admits a probability density function $f(y)$, the expected value $E[Y]$ can be computed as

$$E[Y] = \int_{-\infty}^{\infty} y f(y) dy$$ (5.1)

Therefore, the expected inconsistency time $E[L]$ is

$$E[L] = E[t_0 + r - X]$$
$$= E[r - \gamma]$$
$$= \int_{0}^{\infty} (r - \gamma)\lambda e^{-\lambda \gamma} d\gamma$$
$$= \int_{0}^{r} (r - \gamma)\lambda e^{-\lambda \gamma} d\gamma$$
$$= r + \frac{e^{-r \lambda} - 1}{\lambda}$$
$$= \varphi(r, \lambda)$$ (5.2)
The expected inconsistency ratio, which is defined as the fraction of inconsistency time, is

\[
\phi(r, \lambda) = \frac{\varphi(r, \lambda)}{r} = 1 + \frac{e^{-r\lambda} - 1}{r\lambda}
\] (5.3)

As shown in Fig 5.2, state consistency (i.e. \(1 - \phi\)) drops as expected when the refresh interval \(r\) increases. However, the amount of the decrease depends on the state change rate \(\lambda\). For example, when the state change rate is relatively low (i.e. \(\lambda = 0.05\)), the consistency reduces gradually with the increase of refresh interval. In addition, the maximum inconsistency ratio is moderate, 57% in this case. On the other hand, when the state change rate is relatively high (i.e. \(\lambda = 0.5\) or 1), the consistency drops sharply to 20% when the refresh interval increases from 1s to 4s; after that, the consistency ratio levels out and increasing refresh intervals does not have significant impact on the performance.

In summary, the impact of refresh interval on consistency largely depends on the state change rate; under frequent state changes, tuning state update interval does not have much impact on state consistency.

### 5.3.2 Soft State Model with Channel Loss

Let \(Y\) be the time of first failure occurrence after the last state refresh.

\[P(Y > t) = e^{-\lambda t}\]

For a refresh interval with length \(S\), the expected inconsistency time (or failure recovery time) is

\[E[S - Y] = g(s)\]

Among \(n\) refresh intervals, the total inconsistency time is \(ng(s)\).
Let $p$ be the channel loss probability. With channel loss, the length of the refresh interval observed at the other end of the channel could be $r$, $2r$, ..., $kr$, subject to certain probability. Let the random variable $S$ be the length of a refresh interval.

The probability of successful refresh on each trial is $1 - p$. The probability that $k$ trials are needed to get one successful refresh is ($k = 1, 2, 3, ...$)

\[
P(S = r) = 1 - p \\
P(S = 2r) = p(1 - p) \\
... \\
P(S = kr) = p^{k-1}(1 - p)
\] (5.4)

Then according to the Geometric distribution density function,

\[
E[S] = \frac{r}{1 - p}
\]

\[
E[S - Y] = E[\phi(S)] \\
= \sum (\phi(kr)p^{k-1}(1 - p)) \\
= \phi(r)(1 - p) + \phi(2r)p(1 - p) \\
+ \phi(kr)p^{k-1}(1 - p) + ... \\
= \frac{r}{1 - p} - \frac{e^{\lambda r} - 1}{\lambda(e^{\lambda r} - p)}
\] (5.5)

The expected inconsistency ratio is

\[
\frac{ng(s)}{nE[s]} = \frac{g(s)}{E[s]}
\]

Therefore, the expected inconsistency ratio with channel loss is

\[
\phi(r, \lambda, p) = \frac{r}{1 - p} - \frac{e^{\lambda r} - 1}{\lambda(e^{\lambda r} - p)} \\
= 1 - \frac{e^{\lambda r} - 1}{\lambda(e^{\lambda r} - p)} \frac{1 - p}{\lambda r}
\] (5.6)

From Fig 5.4 we can reach similar analytic results on the impact of refresh intervals on system consistency. The impact of refresh interval on consistency largely depends on state change rate; under frequent state changes, tuning state update interval does not have much impact on state consistency.

5.3.3 Analysis

It is clear that route consistency is closely correlated with routing performance. Route inconsistency between nodes leads to data packet drops due to route loops and route unavailability. Regaining route
consistency quickly helps to reduce data packet drops and improves routing performance. Existing studies on route dynamics [52, 97] have shown that route change rate in mobile ad hoc networks is relatively high, typically, $\lambda \geq 0.5$. On the other hand, recent theoretical analysis of link dynamics [98] indicates that, under reasonable node velocity (i.e. $v \leq 20m/s$), the expected link lifetime is larger than 20s, that is, $\lambda \leq 0.05$.

Consider an arbitrary period $[t_0, t_0 + \delta t]$. Let $X_R$ be the time of first route change occurrence after $t_0$ and $X_i(i = 0, 1, 2...k)$ be the time of first link change occurrence after $t_0$. Assume each link establishment event leads to a new route. Since the route is re-selected when one of its links get broken, or a shorter path between the end nodes appears,

$$X_R = \min(X_i)$$

Therefore,

$$\lambda_R(t_0) = \max(\lambda_i(t_0))$$

The expected route change rate is thus far larger than the expected link change rate.

Therefore, we can infer that tuning HELLO intervals of proactive MANET routing protocols would have significant impact on routing performance while tuning TC intervals would not.

5.4 Simulation

5.4.1 Set-up

We used the OLSR & DSDV implementation that runs in version 2.9 of NS2 [90] and the ad-hoc networking extensions provided by CMU [91], with a radio range of 250m and IEEE MAC/802.11 as the media access control. The detailed configuration is shown in Table 5.2.

In the experiments, we use a network consisting of $n$ nodes: $n = 20$ to simulate a low-density network, $n = 50$ to simulate a high-density network. All simulations run for 100s.

We employed the Random Trip Mobility Model, "a generic mobility model that generalises random waypoint and random walk to realistic scenarios" [83] and performs perfect initialisation. Unlike other
Table 5.2: MAC/PHY Layer Configurations

<table>
<thead>
<tr>
<th>MAC Protocol</th>
<th>IEEE 802.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Radio Propagation Type</td>
<td>TwoRayGround</td>
</tr>
<tr>
<td>Interface Queue Type</td>
<td>DropTailPriQueu</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>OmniAntenna</td>
</tr>
<tr>
<td>Radio Radius</td>
<td>250m</td>
</tr>
<tr>
<td>Channel Capacity</td>
<td>2Mbits</td>
</tr>
<tr>
<td>Interface Queue Length</td>
<td>50</td>
</tr>
</tbody>
</table>

random mobility models, Random Trip reaches a steady-state distribution without a long transient phase and there is no need to discard initial sets of observations.

The average node speed, \(v\), ranges between 1m/s to 30m/s. For example, when the mean node speed is 20m/s the individual node speeds are uniformly distributed between 0m/s and 40m/s. The average node pause time is set to 5s.

A random distributed CBR (Constant Bit Rate) traffic model is used. This allows every node in the network to be a potential traffic source and destination. The rate of each CBR traffic flow is 10kb/s. The CBR packet size is fixed at 512 bytes. There are at least \(n/2\) data flows that cover almost every node.

For each sample point presented, 100 random mobility scenarios are generated. The simulation results are therefore statistically presented with the mean of the metrics and the errors. This reduces the chances that the observations are dominated by a certain scenario that favours one protocol over another.

5.4.2 Scenarios

In our simulations, we use two types of MANET application scenarios: synthetic scenarios and concrete scenarios. Synthetic scenarios provide ideal test environments and examine the signalling performance under a variety of conditions, especially for emerging application. Concrete scenarios provide realistic test fields under certain conditions.

For the synthetic scenarios, nodes are placed in an area of 1000m by 1000m. The node number ranges from 20 to 50. The average node speed ranges from 1m/s to 30m/s. The transmission range is from 50 meters to 400 meters.

Two concrete scenarios are proposed in this study.

- **Urban Scenario.** This scenario describes police security operation with a total of 20 nodes in a field of 1200m by 1200m in urban area. The scenario consists of policemen and police cars. Each policeman walks within the area with the average speed ranging between 1-2 m/s, while each police car runs along the street with the movement speed ranging from 1m/s to 30m/s. The pause time is between 0-5 seconds. The mobility model for both of them is a restricted random waypoint implementation of the Random Trip model. A real-world road map of Houston is used, consisting of residential areas with street intersections. In particular, the area has 594 roads with 383 intersections.
5.5. Observations

- **Rescue Scenario.** This scenario is that of a disaster-rescue operation with a total of 50 nodes in an area of 1000m by 1000m. It consists of 2 helicopters, 2 rescue teams of soldiers by foot and 2 teams on vehicles. The helicopters move with speeds ranging between 1-30 m/s according to the random trip model. The first vehicle team consisted of 10 nodes while the second team consisted of 8 nodes. Both vehicle teams moved according to the random trip model with speeds ranging between 1-20 m/s. The two teams by foot consisted of 15 nodes each, moving with speeds ranging between 1-5 m/s and pause times between 0-5 seconds. Each team covered well-defined areas within the field with sufficient overlap to ensure that information could be relayed among the different teams.

5.4.3 Metrics

In every simulation, we measure the throughput and control traffic overhead of each CBR flow and use them to calculate the mean performance of each metric.

*Throughput* is computed as the amount of data transferred (in bytes) divided by the simulated data transfer time (the time interval from sending the first CBR packet to receiving the last CBR packet). Since we keep the data rate constant, the throughput in this study could represent the end-to-end packet delivery ratio.

The control overhead consists of HELLO messages and TC messages. Considering the broadcasting nature of the control message delivery, the control packets are counted by summing the size of all the control packets *received* by each node during the whole simulation period.

5.5 Observations

In this section, we present the observations on the routing performance under various factors, such as node velocity, node density and refresh intervals.

5.5.1 Observations under Synthetic Scenarios - OLSR

In Fig 5.5, we hold the topology update (TC) interval ($t$) and the radio range ($rr$) constant, and in Fig 5.6, we hold the neighbour update (HELLO) interval ($h$) and the radio range ($rr$) constant.

**Throughput**

From Fig 5.5 we observe that increasing HELLO intervals has a significant impact on throughput. In particular, when node mobility is relatively low ($v = 1 m/s$), the relationship can be approximated with a linear function. When node mobility is relatively high ($v > 5 m/s$), the throughput is approximately proportional to the inverse of the refresh intervals.

These observations concur with the analytic results in Section 5.3. The increase in node velocity leads to an increase in the link change rate. As discussed in Section 5.3, when the link change rate is relatively high, the consistency drops faster than under low mobility, as the HELLO interval is increased from 1s to 5s. When the HELLO interval is larger than 5s, the average throughput converges towards a constant, and increasing refresh intervals does not have significant impact on the performance.

From Fig 5.6 we can see that increasing the TC intervals has no significant impact on routing per-
5.5. Observations

Figure 5.5: Impact of HELLO Intervals on OLSR Throughput \((t=5 \ rr=250)\)

Figure 5.6: Impact of TC Intervals on OLSR Throughput \((h=2 \ rr=250)\)

Figure 5.7: Impact of Refresh Intervals on OLSR Throughput \((n=20 \ rr=125)\)
formance. This provides further support for the analytic results. Since the route change rate is relatively high ($\lambda \geq 0.5$) (see Appendix B), a reduction in TC intervals only has slight impact on performance. However, a small TC interval might introduce excessive control overhead, which leads to channel congestion and throughput degradation as shown in Fig 5.6(b).

As shown in Fig 5.7, the same phenomena is observed when the radio radius is adjusted to 125m.

Traffic Overhead

Fig 5.8 and Fig 5.9 show that increasing refresh intervals impacts on the traffic overhead. According to existing studies of HELLO overheads and topology flooding overheads [43], both HELLO messages and topology control messages generate $O(n^2)$ control overhead.

As shown in Fig 5.8 and Fig 5.9, the traffic overhead drops with the increase of refresh intervals, which can be approximated by inversely proportional relationship. However, a comparison of control overhead reduction shows a different impact. For example, in low density networks (Fig5.8(a) and
5.5. Observations

Fig 5.9(a)), adjusting TC intervals from 1s to 5s leads to 2.5 times overhead reduction than adjusting HELLO intervals. On the other hand, in high-density networks (Fig 5.8(b) and Fig 5.9(b)), adjusting TC intervals from 1s to 5s brings 4 times more overhead reduction than adjusting HELLO intervals.

In summary, due to their broadcast nature, topology control messages generate more overhead than HELLO messages. Consequently, increasing TC intervals has a more significant impact in reducing control overhead than increasing HELLO intervals.

5.5.2 Observations under Concrete Scenarios - OLSR

![Figure 5.10: OLSR Throughput vs. Node Velocity (t=5 rr=250)](image)

(a) Low Density  
(b) High Density

For Fig 5.10, we hold $t$ constant, and for Fig 5.11, we hold $h$ constant.

From Fig 5.10 and Fig 5.11 we see that with the increase of node speed, the average throughput drops almost linearly. Further, by comparing Fig 5.11(a), 5.10(a) with Fig 5.11(b), 5.10(b) we can see that the linearity in high-density networks is better than in low-density networks.

From Fig 5.10, under various scenarios (i.e. node speed and node density), we see that increasing temporal state update rates could improve the average throughput. In addition, by comparing Fig 5.10(a)
and Fig 5.10(b) we can see that the improvements in high-density networks are more significant than in low-density networks.

As seen in Fig 5.10(a), only when the average speed is relatively high \( (v \geq 15\text{m/s}) \) (no overlaps on the error bars) are the average throughputs under different temporal update rates significantly different. On the other hand, from Fig 5.10(b), a significant difference in the average throughputs is observed when the average speed is relatively low \( (\text{i.e. } v \text{ greater than approximately } 7\text{m/s}) \).

Similar patterns are not observed in terms of topological update rates. From Fig 5.11(a), under various scenarios, increasing topological update rates brings no significant improvement to the throughput. The means of the average throughputs with the update interval \( t = 2 \) are only slightly better than those with longer update intervals. In addition, as seen in Fig 5.11(b), in high-density networks, increasing the topological update rates leads to significant performance degradation. This could be explained by the large volume of control traffic, which finally causes congestion and leads to data packet drops.

### 5.5.3 Observations - DSDV

In DSDV, it is not possible to evaluate the impact of temporal updates and topological updates separately. Therefore, we present the observations on the impact of update intervals of routing tables on DSDV routing performance.

![Figure 5.12: Impact of Update Intervals on DSDV Performance \((n=20 \text{ } rr=250)\)](image)

From Fig 5.12 and Fig 5.13 we obtain similar observations on the impact of update intervals on routing performance, which also matches the analytic results well.

As shown in Fig 5.12(a) and Fig 5.13(a), the throughput drops faster under high mobility than under low mobility, as the update interval is increased from 1s to 5s. When the update interval is larger than 5s, the average throughput converges towards a constant, and increasing refresh intervals does not have significant impact on the performance.

As shown in Fig 5.12(b) and Fig 5.13(b), the traffic overhead drops with the increase of update intervals, which can be approximated by inversely proportional relationship.
5.6 Summary

In this chapter, we present a quantitative analysis on the impact of periodic refresh updates for two proactive MANET routing protocols and assess its impact on routing performance. An analytic resilience model is presented based on probability theory, from which we infer that the signalling performance depends largely on various environmental factors, including node mobility. Further simulation results support the analytic results and show that the temporal state updates have a significant impact on the throughput, while the topological state updates do not. This study therefore presents a clear understanding of soft-state signalling performance that gives insightful guidance in designing new applications or network protocols.

Compared with existing studies [30, 33, 31], the approach presented in this chapter combines a model-based analytic study and simulation based performance evaluation study. Without losing any generality, a soft-state model is developed specially for mobile ad hoc networks, with a range of factors such as node velocity, link change rate, route change rate and link loss ratio. We moot that such considerations make our model more accurate in analyzing soft-state performance in presence of node mobility and link dynamics as in mobile ad hoc networks.
Chapter 6

Topology Control and Maintenance for Mobile Ad Hoc Networks Using OLSR

Network dynamics and resource constraints are two of the most critical issues in wireless ad hoc networks and sensor networks. For example, each node running a proactive MANET routing protocol maintains routing information to every other node in the network at all times. Due to the frequent changes caused by mobility, the routing information has to be updated frequently to reflect the topology changes and guarantee the correctness of route selection. This requires the nodes of the proactive routing protocol to broadcast topology updates with efficient strategies. However, the overhead introduced by the topology state advertisements may lead to performance degradation and consume battery power.

Topology control and maintenance algorithms have been proposed to maintain network connectivity while reducing resource consumption (including energy consumption and bandwidth consumption). The key idea of topology control is to define the network topology by forming the proper neighbour relationships under certain constraints. The motivations for topology control/maintenance algorithms are to achieve optimisations in resource usage. However, network performance must be maintained. For example, nodes with certain energy-efficient topology control algorithms could determine their transmission power collaboratively, instead of transmitting using the maximal power. This would reduce energy consumption while maintaining network capacity.

Proactive mobile ad hoc networks use soft-state mechanisms for topology control and maintenance. Topology information is advertised with periodic topology control (TC) messages. Spatially, instead of broadcasting topology changes using simple flooding, proactive routing protocols define topology by carefully selecting the participating nodes in message forwarding. For example, the MPR (Multi-point Relay) mechanism is used in OLSR (Optimised Link State Routing) [16]. Correspondingly, the control traffic under such topology control algorithms is optimised in terms of the amount of topology state (i.e. size of control packets) and the retransmission times of control packets.

Despite the wide deployment of soft-state topology control/maintenance mechanisms, we contend that its impact on routing performance has not yet been well understood. The unique characteristics of MANETs call for a comprehensive and quantitative analysis of performance.

In this chapter, we present an in-depth study on various topology control/maintenance options and
their impacts on routing performance. The efficiency of topology advertisements is investigated from two aspects, namely temporal redundancy (i.e. adjusting the topology update frequency) and spatial redundancy (i.e. using additional topology state information in each topology control message). We compare the routing performance under temporal/spatial redundancy options against that of the original protocol, to show the impacts of the redundancy options. The performance metrics include throughput and control overhead. Detailed definitions of the metrics can be found in Section 6.1.4.

- **Temporal Redundancy.** In this study, we investigate the impacts of temporal topology maintenance strategies. Two types of topology update strategies are studied, namely proactive update strategy and reactive update strategy. The performance of these options is evaluated using statistical analysis, under different scenarios by varying node density and node velocity. A topology update model based on probability theory is used to evaluate the quantitative relationship between topology update intervals and routing performance.

  We find that the proactive topology update intervals have no significant impact on the routing throughput. Further, compared with proactive update strategy, the reactive update strategy in MANET routing protocols does not lead to better throughput while introducing overhead.

- **Spatial Redundancy.** It is known that introducing extra state information in topology update messages increases the control overhead. However, it is not clear whether state redundancy improves the routing performance (i.e. throughput). In this study, we present a quantitative analysis to investigate the impact of spatial redundancy on proactive routing performance. In particular, three options for link state advertisements are studied: the directed MPR subgraph option, the undirected MPR subgraph option and the full graph option.

  Simulation based experiments are performed in order to quantify the impact of several factors on performance. The results of our analysis reveal that in low-density networks, spatial redundancy improves the routing throughput, at the cost of increased control overhead. In high-density networks, spatial redundancy leads to performance degradation.

  Through this study we aim to gain a better understanding of the topology update strategies. The results provide useful insights into how effective flooding-based protocols and applications can be designed in the context of mobile ad hoc networks.

  The rest of this chapter is organised as follows. We present a detailed discussion on a variety of temporal redundancy strategies in Section 6.1, followed by topology spatial redundancy options in Section 6.2. In these sections, we first introduce briefly some background information and identify the major problems, and then propose several topology advertisements options and illustrate our experimental design, including detailed simulation parameters and metrics. At the end of Section 6.1 and 6.2, we present our observations based on NS2 simulations. We finally summary the chapter in Section 6.3.
6.1 Temporal Redundancy

6.1.1 Overview

Each node running the proactive MANET routing protocols maintains routing information to every other node in the network at all times. The routing information can be in the form of either topological repositories (such as OLSR) or distances to other nodes (such as DSDV [23]). Due to frequent topology changes introduced by mobility, the routing information in each node has to be updated to reflect the topology changes and thus guarantee the correctness of route selection. This requires the topology update strategies to be efficient.

Proactive routing protocols use periodic updates to maintain the routing information for each node in the network. For example, OLSR nodes propagate topological control (TC) messages among all the nodes of the network to advertise the link status between themselves and their neighbours. Each node in the OLSR network receives the topology update messages and updates the state repositories correspondingly.

Despite the simplicity and robustness of such a periodic topology update strategy, there have been several concerns about its performance.

- Topology changes may be too dynamic to be captured by periodic updates. Link breakage might have to wait for a period of the topology update interval before being advertised. In presence of frequent topology changes, the performance of the soft-state updates needs to be re-evaluated.

- Although it is commonly believed that a smaller topology update interval (i.e. temporal redundancy) could speed up adaptation to changes, the quantitative impacts of update intervals on routing performance is still not clear. Considering the resource constraints of MANETs, the topology update performance needs to be quantified in order to enhance system efficiency and scalability.

Studies on topology update strategies [63, 65] have been focused on overhead reduction. Samar and Haas propose several update strategies to maximise the update period while maintaining the performance and satisfying certain requirements (such as bounded-delay) [63]. Clausen applies the concept of fisheye routing [64] into OLSR, to reduce the control overhead by introducing temporal partiality into proactive updates [65].

Here we investigate the impact of temporal topology maintenance strategies. We find that the intervals of proactive topology updates have no significant impact on the routing throughput. Compared with the proactive update strategy, adopting the reactive update strategy in the MANET routing protocols does not improve the throughput while increasing the overhead.

Our contributions include,

- We give a quantitative analysis on the impact of topology update intervals on routing performance. We define consistency statistically and use it to evaluate the topology update performance. A novel resilience model is presented for periodic topology updates in mobile ad hoc networks.

- We evaluate the performance of reactive topology updates and proactive updates with a combination of model-based analytic study and simulation based performance evaluation study.
6.1. Temporal Redundancy

6.1.2 Topology Maintenance Strategies in Proactive MANET Routing Protocols

In this section, we first briefly discuss the topology update strategies used by existing MANET routing protocols. We illustrate the details of the topology update strategies investigated in this study.

Existing Topology Update Strategies

From the scope of the update messages, existing topology update strategies can be categorised into global updates and localised updates.

Global Updates. Proactive protocols such as OLSR use global topology updates. In those protocols, each node periodically exchanges its topology information with every other node in the network. The disadvantage of global updates is that they introduce traffic overhead, which consumes a significant amount of bandwidth.

Localised Updates. To reduce the overheads associated with topology updates, in protocols such as DSDV (Destination Sequenced Distance-Vector Routing) [23] and FSR (Fisheye State Routing) [64], each node only propagates route updates within a localised region. For example, a DSDV node maintains the distances (i.e. the number of hops) to all the other nodes in the network and broadcasts periodically such distance information only to its neighbours. The Fisheye State Routing (FSR) [64] adopts the strategy of temporal partiality. Each node running FSR only advertises information about closer nodes and exchanges link state information with its neighbours.

From the triggering mechanism of the topology updates, existing topology update strategies can be categorised into proactive updates and reactive updates.

Proactive Updates. The routing protocols broadcast topology updates periodically, even without topology changes. Some protocols such as OLSR broadcast the updates with a fixed interval. IARP (Intrazone Routing Protocol) [99] and fast-OLSR extension [17] set the update intervals inversely proportional to the maximum velocity of the nodes. TBRPF (Topology Broadcast based on Reverse-Path Forwarding) [24] generates two types of updates: full-topology soft updates and differential updates. DSDV advertises the routing information periodically and incrementally as topological changes are detected.

Reactive Updates. Traditional link-state routing protocols such as OSPF send an update when a link becomes invalid or when a new node joins the network. The benefits of this strategy are that if the network topology and conditions have not changed, no update packets are sent, which eliminates redundant periodic update dissemination into the network. In addition, topology changes can be captured and broadcast quickly. This can reduce data packet drops which may have occurred due to mobility.

Topology Update Strategies Investigated in This Study

In this study, we investigate the impact of the topology update strategies on routing performance. In particular, we study the impact of topology update intervals of a proactive update strategy on routing performance, and evaluate the relative performance of a proactive update strategy and a reactive update strategy. In addition, we propose the following two reactive update options and compare their performance with that of proactive updates:
6.1. Temporal Redundancy

Table 6.1: Model Variables

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau )</td>
<td>Topology update interval</td>
</tr>
<tr>
<td>( \lambda )</td>
<td>Topology change rate</td>
</tr>
<tr>
<td>( \lambda_v )</td>
<td>Topology change rate for a node with velocity ( v )</td>
</tr>
<tr>
<td>( L_t )</td>
<td>Topology inconsistency time</td>
</tr>
<tr>
<td>( \phi_t )</td>
<td>Topology inconsistency ratio</td>
</tr>
<tr>
<td>( \psi )</td>
<td>Derivative of ( \phi_t ) with respect to ( \tau )</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>Control traffic overhead</td>
</tr>
<tr>
<td>( \alpha_{hello} )</td>
<td>Control traffic overhead generated by HELLO messages</td>
</tr>
<tr>
<td>( \alpha_{tc} )</td>
<td>Control traffic overhead generated by TC messages</td>
</tr>
<tr>
<td>( E[X] )</td>
<td>Expected value of a random variable ( X )</td>
</tr>
</tbody>
</table>

- **Localised reactive update (etn1)**. We apply the concept of FSR into traditional reactive topology update strategies. Whenever a link change is detected, the node sends its topology updates to its neighbours only. Correspondingly, the node has a clearer knowledge of the nearer nodes than the farther nodes.

- **Global reactive update (etn2)**. Each node running the proactive routing protocols floods topology updates to every other node in the network whenever a link change is detected.

### 6.1.3 Model Based Analytic Study

A Soft-state Topology Update Model

We assume that the arrival of a topology change event is an independent, identically distributed Poisson process with arrival rate \( \lambda \). This assumption is reasonable, if the node degree is small and the nodes are moving randomly so that the process of route change is totally random [51, 52].

Let \( L_t \) be the topology inconsistency time, i.e. the period from the time a topology change occurs to the time all the nodes in the network have updated their topology repositories. Based on the soft-state model proposed in Chapter 5, the expected topology inconsistency time \( E[L_t] \) is

\[
E[L_t] = \tau + \frac{e^{-\tau \lambda} - 1}{\lambda} = \phi_t(\tau, \lambda)
\]

(6.1)

The expected topology inconsistency ratio (i.e. the ratio of the expected topology inconsistency time to the topology update interval) is

\[
\phi_t(\tau, \lambda) = \frac{e^{-\tau \lambda} - 1}{\tau \lambda} = 1 + \frac{e^{-\tau \lambda} - 1}{\tau \lambda}
\]

(6.2)

Consider the impact of topology update interval \( \tau \) on state inconsistency ratio \( \phi_t \), i.e. \( \frac{d\phi_t}{d\tau} \) the
derivative of $\phi_t$ with respect to $r$.

$$\psi(r, \lambda) = \phi'_t(r)$$

$$= \frac{d\phi_t}{dr}$$

$$= \frac{1}{r^2 \lambda} - \frac{1 + r \lambda}{r^2 \lambda e^{r \lambda}}$$

(6.3)

As shown in Fig 6.1, when topology change rate $\lambda$ is relatively high (i.e. $\lambda > 0.25 s^{-1}$ when $r = 5s$), the increase of topology change rate leads to the decrease of the impact of topology update interval $r$. Especially with larger update intervals (i.e. $r = 5s$ or $7s$), the impact of topology update interval is insignificant (i.e. $\psi < 0.06$).

To summarise, the impact of topology update intervals on consistency depends on state change rate. Under frequent topology changes, tuning topology update intervals does not have much impact on topology consistency.

Control Overhead Analysis

Let us first look at control overhead under the proactive update strategy.

Let $\alpha$ be the amount of control overhead. Let $r$ be the topology update interval. Then

$$\alpha = \alpha_{\text{hello}} + \alpha_{\text{tc}}$$

(6.4)

In this study, we keep the HELLO intervals and node density constant. Therefore,

$$\alpha_{\text{tc}} \propto \frac{1}{r}$$

From Equation (6.4) we can see that the control overhead under the proactive update strategy can be approximated by,

$$\alpha = \frac{\alpha_0^p}{r} + c^p$$

(6.5)

where $\alpha_0^p$ and $c^p$ are constants determined by the size of the network, the size of the control packets and HELLO intervals.

The control overhead under reactive update strategy is analysed as follows.
6.1. Temporal Redundancy

We assume that the link change inter-arrival time distribution can be approximated by an exponential distribution with fairly high accuracy [98]. The link change inter-arrival time density function can then be expressed as,

$$f(t) = \lambda_v e^{-\lambda_v t}$$  \hspace{1cm} (6.6)

The average link change interarrival time is $\frac{1}{\lambda_v}$. Correspondingly, the average reactive update interval is inversely proportional to $\lambda_v$.

Then the control overhead can be approximated by,

$$\alpha = \alpha_0 \lambda_v + c'$$  \hspace{1cm} (6.7)

where $\alpha_0$ and $c'$ are constants determined by the size of the network, the size of the control packets and HELLO intervals.

With the increase of node velocity, the link change interarrival time decreases and the link change rate increases [98]. Therefore, under reactive update options, the average update interval decreases and the topology control overhead increases linearly with link change rate.

![Figure 6.2: Control Overhead under Proactive and Reactive Options](image)

Summary of Analytic Study

Our recent study on route duration statistics (Appendix B) has shown that, in a mobile ad hoc network with moderate or high mobility, the route duration could be approximated by an exponential distribution with appropriate parameters. In particular, most of the route duration (i.e. more than 75%) is smaller than 4s. Correspondingly, the topology change rate is larger than $0.25 s^{-1}$.

Based on the analytic results in this section we can infer that, due to frequent topology changes,

- Topology refresh intervals would not have significant impact on routing throughput.
- Reactive update options would introduce more traffic overhead than proactive options in a network with frequent link changes (i.e. $\lambda_v > \lambda_0$ as shown in Fig 6.2).
6.1.4 Simulation Based Performance Evaluation

In the following paragraphs, we briefly introduce the parameters used in the simulations.

Simulation Configuration

We implement the proposed temporal redundancy options in the OLSR implementation, which runs in version 2.9 of NS2 [90] and uses the ad-hoc networking extensions provided by CMU [91], with a radio range ($rr$) of 250m radius and the use of MAC/IEEE802.11 as the media access control. By default, each wireless node has 50 packet buffer spaces and its raw radio link capacity is 2M bps.

The nodes are randomly placed in an area of 1000m by 1000m. We use a network consisting of $n$ nodes: $n = 20$ to simulate a low-density network, $n = 50$ to simulate a high-density network. To illustrate, if the nodes are evenly distributed, in a low-density network, each node has 4 nodes in its transmission range (Fig 6.3(a)), while in a high-density network, each node has 12 nodes in its transmission range (Fig 6.3(b)).

![Low-Density Network vs. High-Density Network (rr=250)](image)

We use the Random Trip Mobility Model, "... a generic mobility model that generalizes random waypoint and random walk to realistic scenarios ..." [83], which performs "perfect" initialisation. Unlike other random mobility models, Random Trip reaches a steady-state distribution without a long transient phase and there is no need to discard initial sets of observations.

All simulations run for 100s. The steady-state simulations enable us to collect the results within a short simulation period (i.e. 100s), which could represent long-run averages (such as 1000s). On the other hand, it is enough to collect mobility data within 100s, since the nodes at moderate or high velocity ($v > 10m/s$) could travel from one edge of the area to the opposite edge within 100s.

The average node speed, $v$, ranges between 1m/s to 30m/s. For example, when the average node speed is 20m/s the individual node speeds are uniformly distributed between 0m/s and 40m/s. The average node pause time is set to 5s.

The traffic model consists of randomly distributed CBR flows. We allow every node in the network to be a potential traffic source and destination. The CBR packet size is fixed at 512 bytes. There are at
least \( n/2 \) data flows that cover almost every node.

The configuration is summarised in Table 6.2.

### Metrics

In every simulation, we measure the throughput of each CBR flow and the control traffic overhead and then use the simulation results to calculate the average performance of each metric.

*Throughput* is computed as the amount of data transferred (in bytes) divided by the simulated data transfer time (the time interval from sending the first CBR packet to receiving the last CBR packet). Since we keep the data rate constant, the throughput in this study could represent the end-to-end packet delivery ratio.

In order to gauge the topology control overhead, we measure both the number of routing messages, including HELLO messages and TC messages, and the number of bytes in the routing packets transmitted. Since we keep the HELLO intervals constant, the *overall* routing overhead could represent TC message overhead. Considering the broadcast nature of the control message delivery, the packets are counted by summing the size of all the control packets *received* by each node during the whole simulation period.

In order to gain good confidence in the measurement results, we run the simulations 100 times for each data point, with different mobility pattern files, i.e. different starting states for the node positions.

### 6.1.5 Observations

In this section, we present the observations from the simulations on the performance of *temporal redundancy* options under various values of the parameters such as node density and node velocity. We hold the neighbour update (HELLO) interval \((h)\) and the radio range \((rr)\) constant.

#### Routing Performance under Proactive Topology Updates

First, we study the impact of topology update intervals on the routing performance.

From Fig 6.4(a) we can see that, in a low-density network, tuning topology update intervals within a certain range (e.g. \( 1 \leq t \leq 20 \)) has no significant impact on routing throughput. And even if we increase the topology update intervals further, the average throughput only drops slightly.

When the node mobility is low (e.g. \( v = 1m/s \)), the throughput is almost *constant* within the refresh interval range between 1s and 10s. When the node velocity is moderate (e.g. \( v = 5m/s \)) or
1. Temporal Redundancy

Figure 6.4: OLSR Throughput vs. Topology Update Intervals ($h=2 \text{ } rr=250$)

Figure 6.5: OLSR Control Overhead vs. Topology Update Intervals ($h=2 \text{ } rr=250$)

Relatively high (e.g., $v = 20m/s$), the increase of the refresh intervals from 1s to 10s only leads to a performance degradation less than 5%.

In a high-density network, however, our simulation results show that changes to the topology update intervals have a more significant impact on the average throughput.

From Fig 6.4(b) we can see that when the refresh intervals are relatively small ($1 \leq t \leq 5$), reduction in the topology update intervals leads to a performance degradation of up to 50%. When the refresh intervals are larger than 10s, the average throughput drops gradually as the topology update intervals are increased. This can be explained by the fact that small topology update intervals generate a large volume of control packets. This can also be observed in Fig 6.5(b).

From Fig 6.5, the control overhead is inversely proportional to the topology update intervals. This matches Equation (6.5) well. When the topology update interval is increased from 1s to 5s, the control overhead decreases by 81%. More specifically, the overhead reduction in high-density networks is 18 times the reduction in low-density networks.
6.2 Spatial Redundancy

Routing Performance under Reactive Topology Updates

![Graphs](image)

Figure 6.6: OLSR Routing Performance under Reactive Topology Update Options (h=2 rr=250)

In this section, we compare routing performance using the proactive update strategy and the reactive update strategy respectively. We observe that, among the three topology update options, the soft-state based proactive option significantly outperforms the two reactive options. The global reactive update option (i.e. etn2) performs slightly better than the proactive approach (Fig 6.6(a)), but it introduces three times the amount of control overhead. Although the localised reactive update strategy (i.e. etn1) introduces much less overhead, its throughput is far from satisfactory.

6.2 Spatial Redundancy

6.2.1 Overview

Nodes running proactive routing protocols broadcast periodic Topology Control (TC) messages in the network to enable other nodes to maintain correct routing information. Due to the limitations on bandwidth and energy, the dissemination of topology advertisements needs to be optimised in order to reduce resource consumption. For example, OLSR uses a Multi-point Relay (MPR) [40] mechanism to reduce the amount of state information in each TC message [16], the number of TC messages generated and the number of TC message retransmissions.

Existing studies of the performance of the MPR mechanism prove its validity in delivering the messages to each node [41], and its efficiency [40, 42]. However, it is not clear how the optimised topology advertisement method and routing changes (in the control plane) may impact data transfer performance metrics (in the user plane) like end-to-end throughput. More generally, it is necessary to clarify how state redundancy (i.e. extra link state information in each TC message) affects the routing and data transfer performance. Although it is known that redundancy in state advertisements introduces extra control overhead, the possible improvements on routing performance and data transfer performance are not well studied.

Clausen et al [100] investigated two topology advertisement options, namely (1) MPR full link-state
option (i.e. only MPR nodes advertise links to all their neighbouring nodes) and (2) full link-state option (i.e. all nodes advertise the links to all their neighbouring nodes). They conclude that the additional link state information provides better robustness against moderate node mobility.

Below, we present an in-depth analysis of the impact of spatial topology redundancy (i.e. state redundancy) on proactive routing performance under various factors, including node velocity, node density and refresh intervals. Three types of topology information are considered: (1) partial topology based on a directed MPR subgraph, (2) partial topology based on an undirected MPR subgraph and (3) full topology consisting of all symmetric links in the network.

### 6.2.2 Protocol Overview

OLSR uses proactive methods to maintain the routing information about each node in the OLSR network. OLSR nodes detect neighbour changes by exchanging periodic HELLO messages, and propagate TC messages among all the nodes of the network to discover available routes in the presence of mobility and resource constraints.

OLSR also inherits the use of the link-state algorithm (LS), using shortest-path-first (SPF) forwarding. Unlike other LS protocols, OLSR optimises flooding via MPRs.

In the following paragraphs, we introduce briefly the topology diffusion mechanisms in the OLSR routing protocol.

OLSR inherits the concept of link state (LS) routing but with flooding optimisations. In traditional LS-based routing protocols, each node sends its local link-state information to its adjacent nodes once it detects the link changes between itself and its neighbours, and the adjacent nodes then forward the information to their neighbours. Unlike the traditional LS method, OLSR uses MPRs [40, 41, 42] to optimise the message flooding.

Each node selects a set of its neighbour nodes as MPRs. A node, which has selected its neighbour A as its MPR, is called the MPR Selector [16] of node A.

The selective flooding based on MPR is efficient in terms of control message delivery. In [41] it is shown that, such flooding eventually reaches all the nodes in the graph. Also, for each node pair in the network, the subgraph consisting of the unidirectional MPR links in the network and all adjacent links (of the node pair) contains a shortest path with respect to the original graph.

In particular, the MPR optimisation includes the following three aspects.

Firstly, in OLSR, only the MPR nodes are responsible for forwarding control traffic. This significantly reduces the number of retransmissions required to flood a message to all nodes in the network.

Secondly, the partial link state is advertised in order to provide shortest path routes. In OLSR, only the states from the MPR selector set is advertised in the topology control messages. Although bi-directional, only the unidirectional link states (i.e. the link status from the MPR nodes towards their corresponding MPR selectors) are advertised. From this the nodes eventually obtain a directed MPR subgraph of the whole network topology. The motivation for such partial state advertisement is to reduce the size of the topology control messages.

Thirdly, only MPR nodes generate the topology control messages (TC), since the MPR selector
set in non-MPR nodes is NULL. This reduces the number of the topology messages generated in the network.

With all the optimisations above, the MPR mechanism provides an efficient method for flooding control traffic by reducing the number of transmissions required and the amount of control traffic flooded. Further details of OLSR and MPR can be found in [16, 40, 41, 42].

6.2.3 Proposed Spatial Redundancy Options

In this section, we list the three topology advertisement options investigated. These options determine the level of each node's knowledge of network topology and the impact on routing performance.

Spatial Redundancy Options for Topology Advertisements

First, we introduce some commonly used symbols in this section. In the graph \(G(V, E)\) with vertex set \(V\) and edge set \(E\), let \(M \subseteq V\) be the MPR set of \(G\), i.e. a set of nodes that are selected as MPRs by at least one node in the network. For each node \(u \in V\), let

1. \(N(u) \subseteq V\) be the set of nodes adjacent to \(u\).
2. \(MPR(u) \subseteq N(u)\) be the MPR set of \(u\), i.e. a set of adjacent nodes that \(u\) selects as MPRs.
3. \(MPRSEL(u)\) be the MPR selector set of \(u\), i.e. a set of adjacent nodes that select \(u\) as MPR.

The topology advertisement options are listed as follows.

1. The Partial Topology Advertisement option of Directed MPR subgraphs \((pta.d)\) is the default strategy used in the standard OLSR protocol proposed by [16]. In this option, only the nodes selected as MPRs generate TC messages with the state information of the MPR-selector neighbours. That is, for each node \(u \in M\), the TC messages contain the state information of the directed link \((u, v)\) for each \(v \in MPRSEL(u)\). Therefore, the unidirectional links from the MPR nodes towards the MPR selectors are advertised in each TC message. Each node holds a directed subgraph of the network topology that consists of these unidirectional links.

2. The Partial Topology Advertisement option of Undirected MPR subgraphs \((pta.u)\) advertises both the MPR selector set and the MPR set. In this option, the nodes selected as MPRs generate TC messages with the state information of the MPR neighbours and the MPR-selector neighbours. Non-MPR nodes generate TC messages with their MPR state information only. That is, for each node \(u \in M\), the TC messages contain the state information of the directed link \((u, v)\) for each \(v \in MPRSEL(u)\) and the state information of the directed link \((u, v')\) for each \(v' \in MPR(u)\). For each node \(u' \in V - M\), the TC messages contain the state information of the directed link \((u', v')\) for each \(v' \in MPR(u)\). Therefore, the bi-directional links between the MPR nodes and their MPR selectors are advertised. Each node obtains an undirected subgraph of the network topology which consists of these links.

3. The Full Topology option \((fts)\) advertises all symmetric links in the network. In this option, each node advertises all the symmetric links to its neighbours. That is, for each node \(u \in V\), the TC messages contain the state information of the directed link \((u, v)\) for each \(v \in N(u)\) if \((u, v)\) is symmetric. Therefore, each node obtains a full undirected graph of the network topology.
The Impacts of Topology Redundancy on OLSR Routing Performance

By introducing topology redundancy, there are two potential changes to the routing processes of OLSR.

Firstly, it is known that applying topology redundancy introduces extra control message overhead. In terms of the options listed above, the size of control messages of option $ptau$ and $fts$ is likely to result in a larger overhead than that in option $pta_d$. Moreover, more control messages are generated in options $ptau$ and $fts$. The increased control traffic leads to an increase in buffer occupancy and network channel contention, which may cause an increase in packet collisions in the wireless medium. In addition, the increased packet size may cause an increase of packet error rate (PER) for a given bit-error rate (BER).

Since we do not change the MPR-based flooding mechanism, the control messages are still re-transmitted by the MPR nodes only. Therefore, for each control message, the number of re-transmissions is not changed.

Secondly, with topology redundancy, the nodes have more choices in route selection, which improves the route availability. In the presence of route breakage, the nodes can re-establish a new route quickly and thus the packet drops caused by route unavailability can be reduced. The route length (number of router hops) between any two nodes is the same under the different topology options [41] because the nodes always select the shortest paths as routes.

Other network characteristics, such as link duration and link change rates, are unchanged, since the node distribution and mobility is unaffected by (nor dependent upon) the topology advertisements.

6.2.4 Simulation Based Performance Evaluation

We implement the proposed spatial topology redundancy options in the OLSR implementation which runs in version 2.9 of NS2. The parameter configuration of this study is similar with 6.1.4 except the following design.

Design

In this study we aim to evaluate the performance of OLSR under different topology advertisement options as described in the previous sections. In addition to the well-known overhead increase, we focus on the possible impact on overall performance such as end-to-end data throughput, which is considered as the most straightforward metric for MANET routing protocols [92].

The overhead of the state redundancy strategy affects the overall performance in terms of resource consumptions (e.g. bandwidth and battery life). Therefore, in order to present a clear analysis of performance, we use over-provisioned networks in the following simulation configuration.

An over-provisioned network has an abundance of bandwidth resources. The traffic load, including the data traffic and the control traffic, never exceeds the resource constraints. Network congestion loss can therefore be eliminated. In such networks, the impact of routing overhead is not considered the key factor to overall performance.

Consider an example network with 50 nodes using IEEE 802.11 DCF mode as the MAC layer protocol. There are 25 constant bit-rate (CBR) flows. Therefore the maximum number of the flows per link is 25. The CBR data rate is set to be 10 kb/s, and the size of the CBR packets $n_c$ is 512 bytes. The average size of the control packets is 250 bytes. The normalised overhead $\rho$ (i.e. the ratio of the number
of control packets $P_c$ to the number of data packets $P_d$ is 10.

The approximate bandwidth consumed by the control overhead (i.e. $\alpha$) in such a system is,

$$\alpha = P_d \times \rho \times n_c$$

$$= \frac{10000}{512 \times 8} \times 10 \times 250 \times 8$$

$$= 50 \text{ kb/s}$$

And the maximum bandwidth consumption per link is approximately

$$10 \times 25 + 50 = 300 \text{ kb/s}$$

So we set the raw radio link capacity to be 2M bps. The network is therefore over-provisioned. Under such configurations, the effects of the overhead are not considered to have a significant impact on the operational performance of the network.

### 6.2.5 Observations

In this section, we present the observations from the simulations on the performance of spatial redundancy options under various values of parameters such as node density and node velocity, and consider the impact of cross-layer support.

#### Control Overhead

The amount of control overhead is determined by (1) the topology redundancy options ($pta.d$, $pta.u$ or $fts$), (2) the value of the control message refresh intervals (e.g. HELLO intervals $h$ and TC intervals $t$) and (3) the node density.

For Fig 6.7, 6.8 and 6.9, we hold $h$, $t$ and radio range ($rr$) constant, and all figures are obtained with cross layer support.

From Fig 6.7 we can see that the topology redundancy options introduce extra control traffic overhead. As expected, full-topology advertisement generates more overhead than partial-topology advertisement. For example, option $pta.u$ leads to 10\% -20\% increase of bandwidth consumption, while option $fts$ leads to up to 35\% increase.

Fig 6.8 and 6.9 give further insights into the overhead increase from packet number and packet size. From Fig 6.8 we can see that, state redundancy options $pta.u$ and $fts$ generate 14\% -17\% more control packets in low-density networks, while 3\%-13\% in high-density networks. From Fig 6.9 we can see that, the control packet size under option $pta.u$ is almost the same as that under the standard MPR option. In low density networks the average packet size under option $fts$ is just slightly larger than that under the other two options, while the maximum control packet size is increased by 15\%. In high-density networks, the control packet size under option $fts$ is increased by up to 1/3.

Based on these observations we find that, compared to option $pta.d$, the extra overhead of option $pta.u$ is mainly due to the increase in the number of control packets, since the size of the control packets is almost unchanged. On the other hand, the overhead increase of option $fts$ is from both sources.
6.2. Spatial Redundancy

![Graphs](image)

Figure 6.7: Amount of OLSR Control Overhead under \( pta.d \), \( pta.u \) and \( ft.s \) \((h=1 \ t=5 \ rr=250)\)

Figure 6.8: Number of OLSR Control Packets under \( pta.d \), \( pta.u \) and \( ft.s \) \((h=1 \ t=5 \ rr=250)\)

In addition, compared with option \( pta.u \), the extra overhead of option \( ft.s \) is mainly from the increase in control packet size, since option \( pta.u \) propagates almost the same number of control packets as option \( ft.s \).

Furthermore, by comparing Fig 6.7(a) 6.8(a) 6.9(a) with Fig 6.7(b) 6.8(b) 6.9(b) we find that, option \( ft.s \) is more sensitive to network size than option \( pta.u \).

Note that, in IEEE 802.11 wireless networks, the bit error rate (BER) can be quite low. For example, a bit error rate of better than \( 10^{-5} \) is considered acceptable in wireless LAN applications. Therefore, with the above increase of control packet size, the packet error rate would not increase significantly. Moreover, with our over-provisioned design, the increased number of the control packets would not lead to increase in channel contention.

So, in this study, the overhead introduced by \( pta.u \) and \( ft.s \) will not lower the performance of the data packet delivery.
6.2. Spatial Redundancy

![Image: Figure 6.9: Size of OLSR Control Packets under $pta.u$ and $ft.s$ ($h=1 \ t=5 \ rr=250$)](image)

(a) Low Density ($n=20$)  
(b) High Density ($n=50$)

**Figure 6.9: Size of OLSR Control Packets under $pta.u$ and $ft.s$ ($h=1 \ t=5 \ rr=250$)**

![Image: Figure 6.10: OLSR Throughput under $pta.d$ and $pta.u$ with Cross Layer Support ($h=1 \ t=5 \ rr=250$)](image)

(a) Low Density ($n=20$)  
(b) High Density ($n=50$)

**Figure 6.10: OLSR Throughput under $pta.d$ and $pta.u$ with Cross Layer Support ($h=1 \ t=5 \ rr=250$)**

**Throughput**

Below, we describe the observations of throughputs under different conditions. In particular, we use MAC layer notification to facilitate link failure detection [20]. Fig 6.10 and 6.11 are with MAC layer support, while Fig 6.12 and 6.13 are not.

For all the figures in this section, we hold $h$, $t$, and $rr$ constant.

The observations under MAC layer support are as follows.

From Fig 6.10(a) and 6.11(a), we can see that, in low-density networks, introducing state redundancy improves the throughput. As expected, option $ft.s$ leads to better performance than option $pta.u$. The improvements are significant when the node velocity is relatively high (i.e. $v$ greater than approximately 12m/s).

Surprisingly, from Fig 6.10(b) and 6.11(b), it follows that in high-density networks, the throughput with topology redundancy is lower than that with partial topology information. In addition, option $ft.s$ leads to more degradation than option $pta.u$. Specifically, the performance degradation starts when the node velocity is relatively low (i.e. $v$ greater than approximately 2m/s) and then gets more significant.
6.2. Spatial Redundancy

![Graphs showing OLSR Throughput under different conditions](image)

Figure 6.11: OLSR Throughput under $pta_d$ and $ft_s$ with Cross Layer Support ($h=1$, $t=5$, $rr=250$)

![Graphs showing OLSR Throughput under different conditions](image)

Figure 6.12: OLSR Throughput under $pta_d$ and $pta_u$ without Cross Layer Support ($h=1$, $t=5$, $rr=250$)

with the increase of node velocity.

After removing the MAC layer support from OLSR, the observations are as follows.

From Fig 6.12 and 6.13, we can see that in networks without the cross-layer optimisation, the impact of state redundancy is less significant. In particular, less improvement is observed in low-density networks, while insignificant performance degradation is observed in high-density networks.

Introducing state redundancy in low-density networks can therefore lead to performance improvements due to increased route availability. In high-density networks, however, the inverse phenomenon is observed. State redundancy lowers the performance. In addition, through comparing the performance with and without MAC support, we find that the MAC layer notification mechanism is one of the key factors that contribute to the impact of state redundancy on routing performance, especially for the performance degradation in high-density networks.
6.3. Summary

In this study, we present a quantitative analysis on the impacts of topology strategies on the routing performance of OLSR. Other proactive MANET routing protocols like DSDV [23] do not require any network-wide topology advertisements. Therefore, the proposed topology advertisement strategies are not applicable to these protocols.

- **Temporal Redundancy Options.** Reducing topology update intervals gives a little improvement on the performance of OLSR, but with a significant increase of control overhead.

  A reactive topology update approach, as adopted in traditional link state routing protocols such as OSPF, does not perform as well as a proactive topology update approach. Particularly, due to the frequent topology changes, the global reactive update approach introduces too much control overhead. The localised reactive approach, although introducing the least overhead, has the worst data packet delivery performance. Therefore, the proactive approach is more suitable for topology update with OLSR.

  Both of these observations match our analytic results in Section 6.1.3.

  These findings do not conflict with [63] and [65]. Instead, our results offer in-depth explanations for the results obtained in previous studies. Due to its insignificant impact, the topology update period can be maximised within certain value ranges, without damaging the performance.

- **Spatial Redundancy Options.** The impact of topology state redundancy depends on a range of factors, including node velocity, node density and cross-layer optimisation.

  We find that the impact of redundancy is more significant in moderate or high mobility networks. In particular, in low-density networks with moderate or high mobility, the redundancy of topology information improves performance. In relatively stable networks, there is no obvious improvement observed.

  Moreover, performance degradation is observed when topology redundancy is applied into high-
density networks. Especially, the degradation is significant when full topology information is advertised in high-density networks with cross-layer optimisation.
Chapter 7

Effective Neighbour Sensing Mechanisms for OLSR and DSDV

One advantage of proactive MANET routing protocols is that they tend to provide lower route discovery latency than on-demand protocols because they maintain route information to all the nodes in the network at all time [14, 15, 19, 23]. However, the disadvantage of this strategy is the excessive traffic overhead that is generated because of the need to disseminate frequent HELLO messages and topology control messages. Because the resources in wireless networks are severely constrained, the increased channel contention could lead to network congestion and a significant lowering of the network performance. Therefore, an essential challenge in ad hoc networks is to design routing protocols with high packet delivery ratio and low control traffic overhead.

In this chapter, we propose two optimised neighbour detection schemes aimed at improving the data packet delivery ratio and lowering the control overhead. They are the Dynamic Timer algorithm, an adaptive proactive routing algorithm that varies the frequency of the neighbour detection messages in response to network load and mobility conditions, and the Fast Neighbour Handshake algorithm, a fast neighbour sensing scheme that uses explicit handshake mechanism in neighbour detection. The Dynamic Timer algorithm helps achieve the balance between routing performance and overhead cost, while the Fast Neighbour Handshake algorithm reduces neighbour discovery latency and improves route availability.

We compare the performance of these schemes with that of the original protocol, to show the improvements in routing performance. The performance metrics include throughput and control overhead. Detailed definitions of these metrics can be found in Section 7.3.

The rest of the chapter is organised as follows. We provide a detailed description of the Dynamic Timer algorithm in Section 7.1, followed by the Fast Handshake algorithm in Section 7.2. In these sections, we introduce briefly some background information, identify the major problems, and propose our solutions to these problems. Section 7.3 outlines the simulation design, including simulation parameters and metrics. The results based on NS2 simulations are shown in Section 7.4. We finally summarise the chapter in Section 7.5.
7.1 Dynamic Timer Algorithm

Due to the resource-constrained nature of MANETs, the performance of proactive routing protocols is very sensitive to the balance between fast response and traffic overhead. Although a shorter refresh interval may allow nodes to adapt faster to any network changes, the increase in traffic overhead might cause channel congestion and lower overall network performance.

In order to mitigate the side effects of the soft update control overheads, we propose a Dynamic Timer algorithm to adjust neighbour detection intervals of proactive routing protocols according to node mobility. Essentially, such an adaptive algorithm is feedback based. The protocol's behaviours (i.e. parameters) are tuned according to the status of hosting environments such as node mobility and channel loss rate, in order to achieve better performance with less control overhead. Therefore, there are two major issues: how to sense network changes and how to tune the protocol parameters.

Up till now, there have been several adaptive routing approaches for MANETs. Benzaid et al [17] present an approach that adjusts refresh frequency based on node mobility and the MPR status of its neighbouring nodes. Ramasubramanian et al [18] propose a zone-based hybrid routing algorithm that combines proactive and reactive strategies. Boppana et al [6] propose an adaptive Distance Vector routing algorithm by adopting flexible route update strategies according to conditions. These adaptive approaches have the following potential drawbacks.

Dependency on network measurement. The routing performance of the schemes proposed in [17] and [18] depends primarily on the accuracy of network measurement. It is an open question as to how to get accurate estimates of real-time network/traffic characteristics in practice.

Increased complexity. For example, in [18], the operations in zone maintenance and continuous network monitoring not only introduce extra processing overhead but also increase the complexity in configuration and implementation. The performance of ADV (Adaptive Distance Vector Routing) [6] is determined by constant trigger thresholds, which need to be manually configured.

Unknown performance bounds. For example, in ADV [6], the route update frequency increases quickly with node mobility, which brings larger overheads than periodic updates. Also, since only partial route information is maintained, ADV takes longer for a new connection to find a valid route.

In order to solve these problems, we propose two dynamic timer algorithms to improve neighbour detection, namely DTMIAD (Dynamic Timer Based on Multiplicative Increase Additive Decrease) and DT-ODPU (Dynamic Timer Based on On-Demand Proactive Update). By tuning the value of refresh intervals of soft-state timers automatically, the refresh updates are triggered based on network load and mobility conditions. We have shown through simulations that the proposed dynamic timer algorithm outperforms traditional proactive neighbour detection approaches in OLSR and DSDV.

Compared with existing algorithms, our approach shows the following benefits.

Firstly, the operations of the proposed algorithm are independent of network measurement and node mobility detection. Based on analytic studies on link change rate, we propose a simple method in detecting node mobility.

Secondly, the proposed algorithm is simple in both configuration and implementation. The adapt-
ability process is fully automated with only a few parameters. Enlightened by the feedback based control theory, the proposed algorithm can be implemented incrementally, with no need to make significant changes to the existing protocols.

Note that, the designers of proactive routing protocols have considered that nodes in a network may have different and individually tuneable refresh intervals. Therefore in the header of OLSR HELLO messages, a $V_{time}$ field is set to advertise the validity time of neighbour entries and a $HTime$ field is set to advertise the node's HELLO interval [16]. In DSDV, the validity time of route entries is part of the advertised routing tables [23].

Overall, this chapter makes the following contributions. Firstly, it presents the novel adaptive neighbour detection algorithms that provide performance improvements. Secondly, it proposes a new method in sensing network dynamics, which is independent of network measurement techniques. Thirdly, it introduces a Multiplicative-Increase Additive-Decrease (MIAD) controller to adjust the soft-state refresh rate to the conditions of node mobility and data traffic.

### 7.1.1 Problem Formalisation

In order to achieve the goals discussed in the paragraph above, we must be able to adjust refresh intervals (or refresh rates), so that as the node velocity increases, the throughput degradation is slower than the existing solutions (with fixed timer intervals). In this section, we present a formal definition of adaptive routing problems.

**Definition**

Let $f$ be a refresh rate (or frequency). Let $T(f)$ be the quantitative relationship between refresh rate $f$ and routing throughput $T$ (see Table 7.1). Let $C(f)$ be the quantitative relationship between refresh rate $f$ and routing overhead $C$. Based on the studies on the impact of HELLO refresh rate on routing performance (see Chapter 5), we can infer that, if other factors (such as velocity $v \in \gamma$) are kept constant,

\[
T(f_1) \leq T(f_2) \\
C(f_1) \leq C(f_2)
\]
Therefore, we aim to find a function $G$, satisfying

$$\exists G, \forall f_1, f_2 \in \theta, f_1 \leq f_2,$$

$$f = G(v, \rho, t),$$

(1)

$$\forall v \in \gamma,$$

$$f_1 \leq f(t) \leq f_2$$

$$T_v(f_1) \leq T_v(f)$$

$$C_v(f) \leq C_v(f_2)$$

(2)

$$\forall v_1, v_2 \in \gamma, v_1 \leq v_2,$$

$$\left| T_{v_1}(f) - T_{v_2}(f) \right| \leq \left| T_{v_1}(f_1) - T_{v_2}(f_1) \right|$$

$$\left| T_{v_1}(f) - T_{v_2}(f) \right| \leq \left| T_{v_1}(f_2) - T_{v_2}(f_2) \right|$$

List (1) describes that the throughput of the proposed algorithm should be larger than (or equal to) standard routing protocol with lower refresh rate (i.e. larger interval), and the control overhead of the proposed algorithm should be less than (or equal to) standard routing protocol with higher refresh rate (i.e. smaller interval).

List (2) describes that the proposed algorithm should show better adaptability to node mobility. That is, with the increase in node mobility, the throughput drop of the proposed algorithm should be less than (or equal to) standard routing protocol.

7.1.2 Proposed Dynamic Timer Algorithm

In this study, we improve the periodic update strategies of existing proactive routing protocols by adapting dynamically the refresh rates in response to neighbour changes. The proposed method uses successive control messages in state maintenance and therefore we do not expect the simplicity and robustness of the soft-state mechanism would be affected. On the other hand, the adaptability to mobility helps achieve the suitable trade-off between routing throughput and control overhead. In the following paragraphs, we present the details of our proposed dynamic timer algorithms, namely $DT.MIAD$ (Dynamic Timer Based on Multiplicative Increase Additive Decrease) and $DT.ODPU$ (Dynamic Timer Based on On-Demand Proactive Update).

Dynamic Timer Based on Multiplicative Increase Additive Decrease

The dynamic timer algorithm based on MIAD is inspired by control-theoretic adaptive mechanisms similar to those widely adopted in the Internet, i.e. Additive Increase Multiplicative Decrease (AIMD) of TCP's congestion window, which is used to adjust sending rates in response to network congestions: the sending rate of TCP in congestion avoidance state is controlled by a congestion window that is
7.1. Dynamic Timer Algorithm

Table 7.2: Notation in Link Change Rate Equation

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\varepsilon$</td>
<td>Standard Complete Elliptic Integral of the Second Kind</td>
</tr>
<tr>
<td>$\phi$</td>
<td>Direction of motion (i.e. degree of the angle with the x axis)</td>
</tr>
<tr>
<td>$p(\phi)$</td>
<td>$1 + 3\cos(2\phi)$</td>
</tr>
<tr>
<td>$rr$</td>
<td>Transmission range</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Average density of nodes within a transmission zone</td>
</tr>
<tr>
<td>$b$</td>
<td>Maximum node velocity ($b &gt; 0$)</td>
</tr>
<tr>
<td>$a$</td>
<td>Minimum node velocity ($0 &lt; a &lt; b$)</td>
</tr>
<tr>
<td>$v$</td>
<td>Node velocity ($a &lt; v &lt; b$)</td>
</tr>
</tbody>
</table>

halved for every window of data containing a packet drop, and increased by one packet per window of data acknowledged. Our approach in this algorithm uses a Multiplicative-Increase Additive-Decrease (MIAD) controller to adapt the soft-state refresh rate $f$ to the conditions of node mobility and data traffic.

Briefly, refresh rate $f$ is multiplied by a factor $\alpha$ ($\alpha > 1$) if node mobility or packet drop rate increases, and otherwise decremented by a factor $\beta$. By aggressively increasing $f$ when the packet failure rate and the network change rate increase, the routing algorithm improves link detection performance, which reduces data packet drops and increases link availability. Whenever the link change rate decreases, the routing algorithm lowers the refresh frequency conservatively until it finally reaches a steady state.

Therefore, the key question is, "What is the quantitative relationship between node mobility and the link change rate?" We clarify this issue in the following paragraphs and present the details of the proposed algorithm.

Any change in the set of links of a node may be due either to the arrival of a new link or to the breaking of a currently active link. Thus, the expected link change rate for a node $\psi$ is equal to the sum of the expected new link arrival rate $\eta$ and the expected link breakage rate $\xi$.

Samar and Wicker studied the theoretical quantitative relationship between link change rate $\psi$ and factors including node velocity in [98]. They found that, in a practical ad hoc or sensor network where "the number of neighbours of a node is bounded", the expected rate of link breakages $\xi$ is equal to the expected rate of new link arrivals $\eta$. Therefore, the expected link change rate for a node $\psi$ equals 2 times of the expected new link arrival rate $\eta$.

$$\psi(v) = \eta(v) + \xi(v) = 2\eta(v) \quad (7.1)$$

Equation (7.2) describes the expected new link arrival rate [98]. The notations used are shown in Table 7.2.

$$\eta(v) = \frac{2R\sigma}{\pi b} \left[ \frac{v^2}{4} \int_0^\pi p(\phi) \log\left( \frac{b + \sqrt{b^2 - v^2 \sin^2 \phi}}{v + v \cos \phi} \right) d\phi \right] + b^2 \varepsilon \left( \frac{v}{b} \right) \quad (7.2)$$
Table 7.3: DT_MIAD Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$h$</td>
<td>HELLO interval of node $i$</td>
</tr>
<tr>
<td>$h_0$</td>
<td>Initial HELLO interval of node $i$</td>
</tr>
<tr>
<td>$f$</td>
<td>Refresh rate of node $i$ ($f = \frac{1}{h}$)</td>
</tr>
<tr>
<td>$link_chg_cnt$</td>
<td>Change rate within current refresh period</td>
</tr>
<tr>
<td>$prev_chg_cnt$</td>
<td>Change rate within previous refresh period</td>
</tr>
<tr>
<td>$prev2_chg_cnt$</td>
<td>Change rate within the period before previous</td>
</tr>
<tr>
<td>$\beta$</td>
<td>Additive decrease rate</td>
</tr>
<tr>
<td>$\alpha$</td>
<td>Multiplicative increase rate</td>
</tr>
<tr>
<td>$h_{max}$</td>
<td>Upper limit of refresh interval</td>
</tr>
<tr>
<td>$h_{min}$</td>
<td>Lower limit of refresh interval</td>
</tr>
</tbody>
</table>

The relationship between link change rate and node velocity is shown clearly in the Fig 7.1. Here, the values of the parameters are set to $a = 0\text{m/s}$, $b = 40\text{m/s}$, $R = 250\text{m}$.

![Figure 7.1: Link Change Rate vs. Node Velocity](image)

Within a certain velocity range ($0 < v < 40\text{m/s}$), the expected link change rate increases with the node velocity (i.e. $\psi'_i > 0$). Moreover, the increasing speed of the expected link change rate increases with the node velocity (i.e. $\psi''_i > 0$). Therefore, we can examine the dynamics of link change rate in order to detect any changes of node mobility.

The pseudocode of the proposed algorithm is as shown in Algorithm 1. We use the notation as shown in Table 7.3.

**Dynamic Timer Based on On-Demand Proactive Update**

Dynamic Timer Based on On-Demand Proactive Update (DT_ODPU) is based on the concept of a Finite State Machine (FSM). The status of a node is roughly classified into two states: dynamic and static. When internal link changes are detected ($link\_chg\_cnt > 0$), the node is dynamic; correspondingly, it uses a smaller refresh interval $h_{min}$. Otherwise, the node is static and uses a larger refresh interval.

---

1. The data shown in the Fig 7.1 is obtained from [98]
Algorithm 1 DT_MIAD

Input: \( h_0 < \frac{1}{\beta} \)

\[
\begin{align*}
  f &\leftarrow \frac{1}{h_0} \\
  \text{link.chg.cnt} &\leftarrow 0 \\
  \text{prev.chg.cnt} &\leftarrow 0 \\
  \text{prev2.chg.cnt} &\leftarrow 0 \\
  \text{rest.of.init()}
\end{align*}
\]

\textbf{loop}

\hspace{1em} \text{Propagate.Refresh.Msg()}

\hspace{2em} \textbf{if} \text{link.chg.cnt} > \text{prev.chg.cnt} \textbf{then}

\hspace{3em} \textbf{if} \text{link.chg.cnt} - \text{prev.chg.cnt} > \text{prev.chg.cnt} - \text{prev2.chg.cnt} \textbf{then}

\hspace{4em} f \leftarrow f \times \alpha

\hspace{4em} \textbf{if} f > \frac{1}{h_{\text{min}}} \textbf{then}

\hspace{5em} f \leftarrow \frac{1}{h_{\text{min}}}

\hspace{5em} \textbf{end if}

\hspace{4em} \textbf{end if}

\hspace{2em} \textbf{end if}

\hspace{1em} f \leftarrow f - \beta

\hspace{1em} \textbf{if} f < \frac{1}{h_{\text{max}}} \textbf{then}

\hspace{2em} f \leftarrow \frac{1}{h_{\text{max}}}

\hspace{2em} \textbf{end if}

\hspace{1em} \text{SynchroniseTimerInterval(}\frac{1}{f}\text{)}

\hspace{1em} \text{prev2.chg.cnt} \leftarrow \text{prev.chg.cnt}

\hspace{1em} \text{prev.chg.cnt} \leftarrow \text{link.chg.cnt}

\hspace{1em} \text{link.chg.cnt} \leftarrow 0

\hspace{1em} \text{DELAY(}\frac{1}{f}\text{)}

\hspace{1em} /* \ldots \text{do something else} \ldots */

\textbf{end loop}
7.1. Dynamic Timer Algorithm

In this algorithm, the state update is still proactive since refresh messages are still exchanged periodically; however, the refresh frequency (or refresh interval) is adjusted in on-demand manner.

The pseudocode of the proposed algorithm is as shown in Algorithm 2.

---

**Algorithm 2 DT.ODPU**

**Input:** $0 < h_{min} < h_{max}$

$h \leftarrow h_{min}$

$pre.refresh.time \leftarrow now$

$link.chg.cnt \leftarrow 0$

rest.of.init() 

**loop**

*if* link.chg.cnt $> 0$ *then*

Propagate.Refresh.Msg()

*else if* now $\geq (pre.refresh.time + h_{max})$ *then*

Propagate.Refresh.Msg()

$pre.refresh.time \leftarrow now$

*end if*

link.chg.cnt $\leftarrow 0$

DELAY($h$)

/* ... do something else ... */

*end loop*

---

In the following paragraphs, we analyse the dynamics of refresh intervals under DT.ODPU.

We now assume that the arrival of a link change event is an independent, identically distributed Poisson process with arrival rate $\lambda$. This assumption is reasonable, if the node degree is small and the nodes are moving randomly so that the process of link change is totally random [51, 52].

If $X_t$ is the number of link changes in the interval $(0, t)$, we have

$$P(X_t = 0) = e^{-\lambda t}$$

Let $h_{min}$ be the lower limit of refresh interval and $h_{max}$ be the upper limit. Let $P_{r_i}$ be the probability that the refresh interval equals $r_i$. Thus the expected refresh interval can be approximated by,

$$h = \frac{1}{f} = \frac{1}{\sum f_i P_{r_i}} = \frac{1}{(1-e^{-\lambda h_{min}}) + e^{-\lambda h_{max}}/h_{max}}$$

As shown in Fig 7.2, the expected refresh interval reduces significantly when the link change rate is increasing and converges gradually to its lower limit $h_{min}$. In addition, the larger the difference $h_{max} - h_{min}$ is, the better adaptability to network changes DT.ODPU would have.
7.2. Fast Neighbour Handshake Scheme

Summary

The proposed Dynamic Timer algorithms enable routing protocols to adapt quickly to node mobility. By reducing refresh intervals aggressively, the nodes could detect link breakage and establishment much faster than existing approaches based on fixed timer intervals. On the other hand, through increasing refresh intervals when there is no node mobility, the proposed algorithms could automatically balance the trade-off between performance and control traffic overhead.

7.2 Fast Neighbour Handshake Scheme

Neighbour detection can have a significant impact on the performance of a routing protocol. Our recent studies [101, 102] have shown that the performance of OLSR is very sensitive to neighbour detection intervals. Voorhaen and Blondia [20] analyse the performance of several neighbour detection approaches for OLSR and demonstrate the impact that changes to the neighbour detection scheme can have on the routing performance.

Traditionally, MANET routing protocols [16, 25] use periodic HELLO messages to detect neighbour changes. A node establishes new connections when it receives HELLO message broadcasts from its neighbouring nodes. On the other hand, if a neighbour state entry is not refreshed within a period, the node time-outs that entry in its neighbour repository and assumes the connection has been lost. Such a HELLO based neighbour detection mechanism is simple in implementation and robust in the presence of channel loss. HELLO messages are broadcast locally as best-effort traffic under the assumption that any loss would be recovered from periodic refresh messages. However, there have been concerns regarding the performance of such neighbour detection scheme in the context of dynamic environments such as MANETs.

Detection latency. The HELLO based mechanism has a relatively large delay in neighbour detection. For example, it takes around 3 seconds on average for OLSR nodes to detect established connections [16]. Such latency might lead to unnecessary packet drops due to route unavailability, especially in low-density networks with scarce network connectivity.

Resource waste. Periodic HELLO messages are broadcast even if no link changes occur, which wastes bandwidth and battery life. A smaller HELLO interval increases channel contention and might
lead to congestion.

We therefore propose a fast neighbour detection scheme. Instead of relying on periodic HELLO message, the proposed scheme uses explicit route handshake mechanisms in neighbour detection, which reduces the latency in connection establishment and improves path availability. In particular, we present two handshake options based on OLSR, namely Unicast based handshake (UHS) and Broadcast based handshake (BHS) options. We validate our scheme using simulation for a modified version of OLSR, showing that our proposed scheme improves routing performance.

### 7.2.1 Explicit Handshake Scheme

In this section, we present the neighbour detection scheme based on explicit neighbour handshake. MANET routing protocols like OLSR only use symmetric links in route calculation. The established (physical) connections would not be available for data transfer until identified as symmetric links by the routing protocols. Therefore, the delay in neighbour detection might lead to routing performance degradation.

The neighbour detection latency of HELLO based routing protocols is caused by the periodic nature of HELLO messages. After receiving the first HELLO message from a neighbouring node, the OLSR node does not respond until it broadcasts the next HELLO message. Essentially, the neighbour handshake process is done implicitly through exchanging periodic HELLO messages.

In our scheme, we use explicit handshake messages to facilitate connectivity detection. More specifically, in addition to periodic HELLO messages, the node sends explicit handshake messages to its neighbours. The basic process is described as follows.

1. Each node broadcasts periodic HELLO messages to its neighbours.

2. When node $A$ receives the first HELLO message from an unknown neighbour $B$, it creates a new entry for directed link ($B \rightarrow A$), and responds instantly with an ACK message to node $B$, containing the status of the new link ($B \rightarrow A$).

3. When node $B$ receives the ACK message, it infers the existence of a bi-directional link ($B \leftrightarrow A$); then node $B$ sends instantly an ACK message to node $A$, containing the status of the symmetric link between them.

4. If, for any reason, the ACK message from $A$ is lost, the following periodic HELLO messages should cover the loss and complete the connection detection process, as it does in existing proactive neighbour detection schemes.

5. Similarly, if the ACK message from $B$ to $A$ is dropped, the following periodic HELLO message will cover the loss and inform $A$ about the symmetric link status, as it does in existing proactive neighbour detection schemes.

**Handshake Options**

We propose two handshake options based on the above scheme, namely Unicast based handshake (UHS) and Broadcast based handshake (BHS).
• **Unicast based handshake.** The handshake packets are transmitted as unicast packets between the neighbouring nodes. For example, when node $A$ receives its first HELLO message from $B$, it only sends ACK messages to node $B$. Other neighbouring nodes of $A$ would may receive the handshake packets but the packet is not addressed to them.

• **Broadcast based handshake.** The handshake packets are transmitted using broadcast packets. For example, when node $A$ receives its first HELLO message from $B$, it broadcasts ACK messages to each of its neighbours. Other nodes may also observe the packet.

The difference between these two handshake options is that the UHS option helps detect link breakage while the BHS option does not. If the MAC layer fails to disseminate a unicast handshake packet (i.e. the UHS option), the handshake packet is sent back to the network layer. The routing protocol like OLSR uses this failure information as an additional criterion\(^2\) to determine whether the link to the destination node (of the handshake packet) has been lost [19, 20].

However, the emission of the unicast handshake packets requires that the node resolve its neighbour's address. This increases the ARP processing overhead and might lead to ARP queue overflow in large-scale mobile networks. The BHS option does not introduce such processing overhead, because the broadcast handshake packets do not need address resolution.

### 7.2.2 Analysis

In this section, we present an analytical study on the performance of the proposed algorithm. In particular, we compare the link detection latency under HELLO based neighbour detection mechanism with that under the proposed fast neighbour handshake scheme. In the following discussions, we assume that:

1. The arrival of a link establishment event is an independent, identically distributed Poisson process with arrival rate $\lambda$.

2. The delay in packet transmission and processing (i.e. $t_p$) is small enough (compared with link detection latency) to be ignored.

These assumptions are reasonable as long as the node degree is small and the nodes are moving randomly so that the process of link establishment is random [51, 52].

**HELLO Based Neighbour Detection**

Let $L_p$ be the link detection latency of the proactive neighbour detection mechanism, and let $X_p$ be the time when the first symmetric link is established. The HELLO interval is $r$.

The link discovery latency can then be expressed as,

$$L_p = t_0 + \Delta_t - X_p + r = r + \Delta_t - (X_p - t_0)$$  \hspace{1cm} (7.3)

According to our assumptions,

$$P(X_p - t_0 > t) = e^{-\lambda t}$$  \hspace{1cm} (7.4)

\(^2\)OLSР mainly uses a time-out mechanism to detect link breakage (see Chapter 3).
### Table 7.4: Notation in Neighbour Detection Analysis

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r$</td>
<td>HELLO interval</td>
</tr>
<tr>
<td>$\lambda$</td>
<td>Link change rate</td>
</tr>
<tr>
<td>$\Delta t$</td>
<td>Time gap between successive messages</td>
</tr>
<tr>
<td>$L_p$</td>
<td>Detection latency of HELLO based method</td>
</tr>
<tr>
<td>$l_p$</td>
<td>Expected detection latency of HELLO based method</td>
</tr>
<tr>
<td>$L_h$</td>
<td>Detection latency of handshake based method</td>
</tr>
<tr>
<td>$l_h$</td>
<td>Expected detection latency of handshake based method</td>
</tr>
<tr>
<td>$t_p$</td>
<td>Latency in packet transmission and processing</td>
</tr>
<tr>
<td>$E[X]$</td>
<td>Expected value of a random variable $X$</td>
</tr>
</tbody>
</table>

### Figure 7.3: HELLO Based Neighbour Detection Process
Therefore, the expected link discovery latency can be approximated by,

\[ l_p = E[L_p] = r + \Delta t - E[X_p - t_0] = r + \Delta t - \frac{1}{\lambda} \]  (7.5)

For example, if the average new link arrival rate is one link per second (i.e. \( \lambda = 1 \)) and the HELLO interval \( r \) is by default 2s, the expected link discovery latency is around 3s.

Proposed Fast Neighbour Handshake Scheme

\[ \text{Figure 7.4: Fast Neighbour Handshake Process} \]

Let \( L_h \) be the link detection latency of the proposed handshake based neighbour detection mechanism, and let \( X_h \) be the time when the first symmetric link is established. If the HELLO interval is \( r \), the link discovery latency under our proposed scheme can be expressed as,

\[ L_h = t_0 + \Delta t - X_h = \Delta t - (X_h - t_0) \]  (7.6)

As before we have,

\[ P(X_h - t_0 > t) = e^{-\lambda t} \]  (7.7)

The expected link discovery latency therefore can be approximated by,

\[ l_h = E[L_h] = \Delta t - E[X_h - t_0] = \Delta t - \frac{1}{\lambda} \]  (7.8)

Factorial Analysis

From the above discussions we can see that the handshake algorithm potentially has a smaller link discovery latency. This provides improved route availability, which in turn leads to better routing performance. In the following paragraphs, we look at the impact of factors, such as node density and node transmission range, on the performance of the proposed handshake algorithm.

Consider the relative improvement in neighbour detection latency (\( \Delta l \)) over the HELLO based scheme.

\[ \Delta l = \frac{E[L_h]}{l_p} = \frac{r}{r + \Delta t - \frac{1}{\lambda}} \]  (7.9)

Equation (7.9) presents a quantitative relationship between the improvement in latency and the factors including the link arrival rate and the refresh interval.
7.2. Fast Neighbour Handshake Scheme

- **Rate of new link arrivals** $\lambda$. Studies [51] on link dynamics show that, the rate of new link arrivals $\lambda$ increases with node velocity $v$, node density $\rho$ and node transmission range $rr$. As we can see in Fig 7.5 that decreasing link arrival rate gives improvements for link detection latency. From this we infer that, in low-density networks with relatively smaller transmission radius, the proposed handshake algorithm is expected to outperform the proactive neighbour detection scheme.

![Figure 7.5: $\Delta t_p$ vs. $\lambda$ ($r=2 \Delta t=1.5$)]

- **Refresh Intervals** $r$. From Equation (7.9) and Fig 7.6, we see that increasing the refresh interval, $r$, improves $\Delta t_p$. Therefore, the proposed handshake algorithm is expected to have a better performance in a network with large refresh intervals $r$.

![Figure 7.6: $\Delta t_p$ vs. $r$ ($\lambda=2 \Delta t=1.5$)]

Summary

The proposed Fast Neighbour Handshake scheme aims to reduce the neighbour discovery latency through the exchange of explicit handshake messages. Therefore, with the proposed scheme, the nodes could detect link establishment faster than a HELLO based neighbour detection approach, leading to fewer data packet drops. As shown in the above analysis, the performance improvement is expected to be significant in low-density networks with small transmission ranges and large refresh intervals. Unfortunately, the performance improvement is at the cost of an increase in the control traffic overhead.
7.3 Performance Evaluation

We have implemented the proposed algorithms into OLSR & DSDV, which run in version 2.9 of NS2 [90] with the ad-hoc networking extensions provided by CMU [91]. The wireless channel type in this study is IEEE 802.11 wireless LAN with distributed coordination function (DCF). The channel has a circular radio range with 250 meters radius and a capacity of 2Mbits. The detailed configurations are shown in Table 7.5, 7.6 and 7.7.

The simulation environments used in this study are similar to those in Chapter 6. The nodes are randomly placed in an area of 1000m by 1000m. We use a network of 20 nodes to simulate a low-density network, and a network of 50 nodes to simulate a high-density network. All simulations run for 100s.

The mobility model used in this study is the Random Trip Mobility Model [83]. The mean node speed, \( v \), ranges between 1m/s to 30m/s. The average node pause time is set to 5s.

Similar to Chapter 6, we use a random distributed CBR (Constant Bit Rate) traffic model, which allows every node in the network to be a potential traffic source and destination. The rate of each CBR traffic flow is 10kb/s. The CBR packet size is fixed at 512 bytes.

We measure the throughput of each CBR flow and the control traffic overhead and then use the

---

**Table 7.5: MAC/PHY Layer Configurations**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC Protocol</td>
<td>IEEE 802.11</td>
</tr>
<tr>
<td>Radio Propagation Type</td>
<td>TwoRayGround</td>
</tr>
<tr>
<td>Interface Queue Type</td>
<td>DropTailPriQueu</td>
</tr>
<tr>
<td>Antenna Model</td>
<td>OmniAntenna</td>
</tr>
<tr>
<td>Radio Radius</td>
<td>250m</td>
</tr>
<tr>
<td>Channel Capacity</td>
<td>2Mbits</td>
</tr>
<tr>
<td>Interface Queue Length</td>
<td>50</td>
</tr>
</tbody>
</table>

**Table 7.6: OLSR Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO Interval</td>
<td>1s</td>
<td>2s</td>
</tr>
<tr>
<td>TC Interval</td>
<td>5s</td>
<td>5s</td>
</tr>
<tr>
<td>Neighbour Hold Interval</td>
<td>3s</td>
<td>6s</td>
</tr>
<tr>
<td>Topology Hold Interval</td>
<td>15s</td>
<td>15s</td>
</tr>
</tbody>
</table>

**Table 7.7: DSDV Parameters**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Route Update Interval</td>
<td>1s</td>
<td>2s</td>
</tr>
<tr>
<td>Route Hold Interval</td>
<td>3s</td>
<td>6s</td>
</tr>
<tr>
<td>Minimum Update Period</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Weighted Setting Time</td>
<td>6s</td>
<td>6s</td>
</tr>
</tbody>
</table>
7.4. Observations

In this section, we compare the routing performance of the proposed algorithms with that of the standard proactive routing protocols, and present the observations under various values of the parameters such as node density and node velocity.

7.4.1 Performance of Dynamic Timer Algorithms

Routing Performance under DT_MIAD

As shown in Fig 7.7 and Fig 7.8, OLSR with DT_MIAD achieves as good performance as standard OLSR with smaller interval (i.e. $h_{\text{LES}}$ in Fig 7.7 and 7.8) but with much less overhead. The overhead of OLSR with DT_MIAD is up to 22.5% less than that of standard OLSR with smaller refresh interval. This demonstrates clearly the benefits of our DT_MIAD algorithm described in Section 7.1.2. The control overhead is reduced because the refresh interval is adjusted automatically to a large value when there are no network changes.

$^3h=1, t=5$
7.4. Observations

![Graph](image)

(a) Throughput

(b) Overhead

Figure 7.8: OLSR Performance under DT_MIAD (n=50, α=2, β=0.02)

![Graph](image)

(a) Throughput

(b) Overhead

Figure 7.9: DSDV Performance under DT_MIAD (n=20, α=2, β=0.1)

![Graph](image)

(a) Throughput

(b) Overhead

Figure 7.10: DSDV Performance under DT_MIAD (n=50, α=2, β=0.1)
7.4. Observations

Compared to standard OLSR with larger interval (i.e. $h_{2t5}$ in Fig 7.7 and 7.8), OLSR with DT_MIA shows good adaptability to node mobility. With the increase in node mobility, the throughput drop of OLSR with DT_MIA is less significant than standard OLSR with larger interval. For example, as shown in Fig 7.8(a), when the node velocity increases from 10m/s to 20m/s, OLSR with DT_MIA has a 14.6% performance drop, while standard OLSR with larger interval has a drop of up to 32.6%. This matches our expectations. When the link change rate increases, the DT_MIA algorithm increases the refresh rate so that state inconsistency can be recovered very quickly after it occurs.

On the other hand, OLSR with DT_MIA has more overhead than standard OLSR with larger interval. As shown in Fig 7.7(b) and Fig 7.8(b), the overhead of OLSR with DT_MIA is 20% - 33% more than standard OLSR with larger interval. Meanwhile, as the nodes move faster, the refresh rate is increased by the DT_MIA algorithm. Therefore, the overhead of OLSR with DT_MIA also increases.

Similar phenomena are observed in the performance of DSDV with DT_MIA, as shown in Fig 7.9 and Fig 7.10. intVal is the sending interval of the node’s route table (see Chapter 3).

To summarise, the DT_MIA algorithm satisfies the requirements described in Section 7.1.1 and matches our expectations described in Section 7.1.2. The simulation results show that, DT_MIA outperforms the standard proactive routing algorithm in terms of the balance of throughput and overhead.

Routing Performance under DT_ODPU

![Figure 7.11: OLSR Performance under DT_ODPU (n=20)](image)

From Fig 7.11 and Fig 7.12, OLSR with DT_ODPU shows as good performance as standard OLSR with a smaller interval (i.e. $h_{1t5}$ in Fig 7.11 and 7.12) but with much less overhead. Especially in low-density networks with low mobility (Fig 7.11(b)), the overhead of OLSR with DT_ODPU is as low as standard OLSR with larger interval (i.e. $h_{2t5}$ in Fig 7.11 and 7.12). With the increase in mobility, the DT_ODPU algorithm increases the refresh rate in response to network changes. Therefore the overhead of OLSR with DT_ODPU increases. However its overhead is still 25% less than standard OLSR with a smaller interval. This can be explained by the fact that the refresh rate is lowered automatically by the DT_ODPU algorithm when there are no network changes.
7.4. Observations

Figure 7.12: OLSR Performance under DT.ODPU (n=50)

Figure 7.13: DSDV Performance under DT.ODPU (n=20)

Figure 7.14: DSDV Performance under DT.ODPU (n=50)
7.4. Observations

Compared with standard OLSR with a larger interval, OLSR with DT.ODPU improves throughput at the expense of increased overhead. With the increase in node mobility, the drop in throughput for OLSR with DT.ODPU is less significant than standard OLSR with larger interval. For example, as shown in Fig 7.12(a), when the node velocity increases from 10m/s to 20m/s, OLSR with DT.ODPU has a 13.7% performance drop, while standard OLSR with a larger interval has a drop of up to 32.6%. This is because the DT.ODPU algorithm increases the refresh rate when the link change occurs and therefore state inconsistency can be recovered very quickly.

On the other hand, OLSR with DT.ODPU introduces more overhead than standard OLSR with a larger interval. As shown in Fig 7.11(b) and Fig 7.12(b), the overhead of OLSR with DT.ODPU is 16% - 23% more than standard OLSR with a larger interval. Meanwhile, as the nodes move faster, the refresh rate is increased by the DT.ODPU algorithm. Therefore, the overhead of OLSR with DT.ODPU also increases.

Similar phenomena are observed in the performance of DSDV with DT.ODPU, as shown in Fig 7.13 and Fig 7.14. intVal is the sending interval of the node's route table (see Chapter 3).

To summarise, compared with standard proactive routing algorithms, DT.ODPU significantly improves the routing performance, while introducing much less control overhead than the traditional method (i.e. improving throughput by reducing refresh intervals). This matches our expectations described in Section 7.1.2.

7.4.2 Performance of Fast Handshake Algorithms

Throughput

![Graphs showing throughput for low and high density networks](image)

Figure 7.15: OLSR Throughput with Fast Handshake Algorithms ($h=2$, $t=5$, $rr=250$)

From Fig 7.15 we can see that, the proposed handshake options outperform the HELLO based option in high-mobility networks. For example, as shown in Fig 7.15(a), OLSR with the UHS option (i.e. olsr+UHS) leads to an 18% increase in throughput, while OLSR with the BHS option (i.e. olsr+BHS) improves it by 12%. With no mobility or low mobility, the occurrence of a link arrival is rare. Therefore, the handshake options have no significant impact on the routing performance.
By comparing Fig 7.15(a) with Fig 7.15(b) we can see that in low-density networks, the proposed
handshake options significantly improve routing throughput. In high density networks, however, the
handshake options give little benefit. This matches our analytical results very well.

In addition, the UHS option outperforms the BHS option in low-density networks. Figure 7.15(b)
shows that in high-density networks, the performance of the unicast option is not quite satisfactory. This
is caused by the increased ARP (Address Resolution Protocol) processing overhead and can be solved
by improving the ARP processing capacity.

Control Overhead

![Graph](image)

Figure 7.16: OLSR Overhead with Fast Handshake Algorithms ($h=2$, $t=5$, $rr=250$)

From Fig 7.16(a) and Fig 7.16(b) we can see that, the UHS option introduces almost no extra control
overhead in low-density networks, while introducing 18% extra overhead in high-density networks. The
BHS option increases the control overhead by up to 36%.

Therefore, with the proposed handshake algorithms, the performance improvements in data packet
delivery are at the expense of increasing control overhead. The control overhead introduced might be
significant, especially in high-density networks with high node mobility.

7.5 Summary

In this chapter, we have proposed two optimised neighbour detection schemes, which aim at improving
the throughput while at the same time keeping the control overhead low. Our simulation results with
OLSR and DSDV show that,

The proposed Dynamic Timer algorithms, $DT.MIAD$ and $DT.ODPU$, have better adaptability and
routing performance than standard proactive routing algorithms. By automatically tuning the value of
HELLO intervals, the refresh rate of HELLO updates is adjusted based on network conditions.

The proposed Fast Neighbour Handshake algorithms improve the routing performance especially
in high-mobility networks. Moreover, the UHS option improves the routing performance significantly in
low-density networks.
Our simulation results reveal one significant drawback of the proposed handshake options: introducing extra traffic overhead in high-density networks with high node mobility. The increased control overhead could cause channel congestion and lower network performance. Therefore, the proposed algorithms are more suitable for networks with abundant resources and high tolerance to data packet loss in terms of packet delivery performance, rather than networks with very tight resource constraints.
Chapter 8

Conclusions and Future Work

We conclude this dissertation with a discussion of our contributions and directions for future work.

8.1 Discussion

The major concerns about proactive MANET routing protocols are route inconsistency due to node mobility and excessive overhead. The causes for these concerns lie in:

(1) Lack of clear understanding of soft-state signalling performance in MANETs. Despite its fundamental importance to proactive MANET routing protocols, the properties of soft-state approach and the circumstances in which it might best be employed are still not fully understood.

(2) Fixed timer intervals. Refresh rates of existing signalling approaches are usually kept unchanged no matter what network conditions (node velocity, node density, etc.) are. Although simple and robust, these approaches do not adequately handle the problems of wireless media, such as the trade-off between performance and the costs.

This dissertation addresses these issues from the aspect of signalling, claiming that effective signalling can improve MANET routing performance without introducing significant control overhead. In particular, the study analyses soft-state signalling performance in existing proactive MANET routing protocols (OLSR and DSDV), and shows the quantitative relationship between routing performance and factors like refresh intervals, node mobility and node density (see Chapter 5). Based on performance analysis, the efficiency of topology control and maintenance strategies is studied to maximise resource usage and alleviate channel contention (see Chapter 6). Optimised neighbour detection schemes are proposed in Chapter 7, either by adapting automatically the value of refresh intervals to node mobility, or using explicit notification messages in establishing link connections.

The relationship between Chapter 5, 6 and 7 is shown in Fig 8.1. Chapter 5 lays the foundation for the studies on topology advertisement strategies in Chapter 6 and neighbour detection schemes in Chapter 7. It gives a mathematical model on soft state performance, and analyses the different impact of temporal update interval and topological update interval. Based on the analytical model and observations of (the impact of) topological update intervals, Chapter 6 presents further investigation into a variety of topology update strategies. Since temporal update (i.e. HELLO) intervals have significant impact on routing performance (Chapter 5), it is possible to propose the optimised neighbour detection algorithms
8.1. Discussion

in Chapter 7.

Performance Evaluation of Soft State Signalling (Chapter 5)

Chapter 5 investigates the impact of soft state timer intervals on routing performance\(^1\). The soft-state model based on probability theory and the experimental study have presented an accurate and comprehensive understanding of MANET signalling performance. It is seen that the impact of soft-state refresh intervals depends on a range of factors, and the intervals of some message types (HELLO messages) have a larger impact on routing performance than other message types. In addition, tuning temporal update rates has a larger impact on routing performance in high-density/high-mobility networks than in low-density/low-mobility networks.

These observations can be explained by the relationship between the impact of refresh intervals (on performance) and link change rate (as shown in Fig 8.2). The link change rate for a node increases with node velocity and node density. Within a reasonable velocity range (i.e. \(0 < v < 30\text{m/s}\)), the average link duration is larger than 10s (see Appendix B), and the average link change rate is smaller than 0.1 \(s^{-1}\). Therefore the impact of refresh intervals increases with (link) change rate when the change rate is smaller than 0.5 \(s^{-1}\).

Increasing topology update rates brings no significant improvement to throughput. This is because the average route change rate within a reasonable velocity range is larger than 2 \(s^{-1}\) (see Appendix B). The impact of refresh intervals is low when the change rate is larger than 2 \(s^{-1}\).

The control overhead drops with the increase of refresh intervals, which can be approximated by an inversely proportional relationship. Because topology update messages are forwarded to each node in the network while temporal update messages are only exchanged locally between neighbouring nodes, topology updates generate more overhead than temporal updates. Consequently, reducing topology update rates has a more beneficial impact on control overhead than reducing temporal update rates.

Topology Advertisement Strategies (Chapter 6)

\(^1\)The impact of time-out intervals has been studied in INSIGNIA signalling [32, 37].
Chapter 6 studies the performance of hybrid topology advertisement strategies. It has been observed that the intervals of proactive topology updates have no significant impact on routing throughput. Therefore topology update intervals can be maximised within a certain value range without damaging the performance. This supports the existing studies of topology update strategies [63, 65].

Reactive update strategies in the studied MANET routing protocol (OLSR) do not lead to improved throughput while producing a large amount of overhead. In particular, the global reactive update approach introduces excessive control overhead. The localised reactive approach, although introducing the least overhead, has the worst packet delivery performance. This leads to the conclusion that proactive strategies are more suitable for MANET topology updates than the reactive strategies.

In addition, introducing state redundancy (i.e., spatial redundancy) in low-density networks leads to performance improvements due to increased route availability. State redundancy has more impact in moderate or high mobility networks. In relatively stable networks, however, no obvious improvement has been observed.

Performance degradation is observed when topology state redundancy is applied to high-density networks. The degradation is particularly strong when full topology information is advertised in high-density networks with cross-layer optimisation. The MAC layer notification mechanism is one of the key contributing factors to the impact of state redundancy on routing performance, especially for performance degradation in high-density networks.

**Effective Neighbour Detection Mechanisms (Chapter 7)**

Chapter 7 looks at the problem of neighbour detection mechanisms, either by adapting automatically the values of refresh intervals in response to node mobility (i.e., the Dynamic Timer scheme), or through the use of explicit notification messages in establishing link connections (i.e., the Fast Handshake scheme).

It has been observed that, the dynamic timer scheme achieves as good routing throughput as standard protocols with small refresh intervals, but with much less overhead. This is because the dynamic timer algorithms aggressively increase the refresh rate when there are link changes, but otherwise reduces the refresh frequency (conservatively or aggressively). As a result, routing protocols with the
8.2. Conclusions

dynamic timer algorithms send frequent refresh messages only when link changes are detected. This leads to significant control overhead reduction.

Compared to the standard protocols with large refresh intervals, the dynamic timer scheme achieves much better throughput at the expense of extra control overhead. However, in low-density networks with low mobility, the dynamic timer scheme has similar overhead to standard protocols. With the increase in mobility, the dynamic timer scheme increases the refresh rate in response to network changes, which leads to an overhead increase. However, the overhead is still much less than that of standard protocols with smaller interval.

The fast handshake scheme reduces the neighbour detection latency. With the handshake options, OLSR nodes detect link establishment faster than the HELLO based approach. This in turn leads to fewer data packet drops and gives a throughput improvement. The performance improvement is more significant in low-density networks with relatively smaller transmission radius, moderate or high mobility, and larger refresh intervals. This is because the link detection improvements are more significant when the link arrival rates are low and the refresh intervals are large.

One drawback of the fast handshake scheme is an increase in control overhead. This can cause channel congestion and lower network performance.

Another limitation of the fast handshake scheme is that it is only suitable for proactive routing protocols that use symmetric links in route calculations. Therefore, the handshake scheme is not evaluated in DSDV since DSDV uses asymmetric links in route calculation.

8.2 Conclusions

This dissertation is focused on the problem of reliable data packet delivery through intermittently connected networks without a priori knowledge of node movements or locations. The key observation of soft-state signalling performance in MANETs is that the impact of soft-state intervals on routing performance depends on node density and node velocity. Reducing refresh intervals may not lead to performance improvements. Temporal state updates on the other hand have a significant impact on throughput, while topological state updates do not.

The investigation of temporal redundancy in relation to topology advertisement strategies shows the relationship between the impact of topology update intervals and route change rate. Compared with the proactive update strategies, the reactive update strategies do not lead to better throughput while introducing increased overhead.

Introducing extra state information in topology update messages increases control overhead. However, its impact on throughput depends on node density and node mobility. In particular, in low-density networks with moderate or high mobility, spatial redundancy improves the performance. In relatively stable networks, it has no obvious impact on throughput. In high-density networks, spatial redundancy leads to performance degradation.

The dynamic timer algorithms achieve as good routing throughput as standard proactive routing protocols with (fixed) small HELLO intervals but with lower routing overhead. It achieves much better throughput than standard proactive routing protocols with (fixed) large HELLO intervals, but at the
expense of some extra control overhead.

The fast handshake scheme reduces neighbour detection latency and improves routing throughput. However, the overhead of the handshake options can cause channel congestion and lower network performance. Therefore, the fast handshake algorithms are more suitable for networks with abundant resources and high tolerance to data packet loss in terms of packet delivery performance, rather than networks with very tight resource constraints.

8.3 Future Directions

There are several interesting directions for future work based on the work described in this dissertation.

- **MANET Variations.** Emerging MANET variations, such as underwater ad hoc networks (UAN), provides interesting scenarios for the work described in this dissertation. For example, a single point to point underwater communication link of UAN suffers from limited radio ranges, especially at higher data rates. Under this scenario, the topology advertisement strategies and the neighbour detection schemes in this dissertation might lead to significant performance improvements.

Another example research area is Vehicle Networks. Extending networking capability to automobiles enables us to build applications such as real-time traffic routing, traffic/congestion alert, remote sensing, and collision avoidance systems. In such ad-hoc inter-vehicle networks, nodes (i.e. vehicles) are highly mobile. Thus, communications must be opportunistic and delay-tolerant. The algorithms in this dissertation would require further improvements in order to support applications in Vehicle Networks.

This dissertation investigates routing performance over wireless LANs (IEEE 802.11), the most popular access technology. In order to support emerging applications in heterogeneous networks, signalling performance in other wireless networks (such as bluetooth networks) needs to be addressed in the future work.

- **Wireless Sensor Networks.** Compared with MANETs, wireless sensor networks (WSNs) have more stringent resource constraints and reliability limits (including processing power, battery lifetime, bandwidth and unreliability of the transmission etc). In order to enable a WSN to support real-time applications and services, one of the key issues is to develop resource-efficient protocols in maintaining network states. Further to our performance studies on topology control and neighbour detection in MANETs, it would be interesting to evaluate quantitatively the signalling performance of existing routing protocols in WSNs, especially in terms of energy efficiency and QoS guarantees.

In order to support emerging applications of WSNs, another fundamental issue is self management. The networks should be self-configuring, so that new network devices are automatically detected and configured without human intervention. In addition, WSNs have to be self-healing in order to detect failures and recover from events that might cause some of its parts to malfunction. Moreover, the proposed solutions should promote the resources productivity without incurring a
8.3. *Future Directions*

high cost to the network. From these aspects, the work in this dissertation may offer avenues for enabling adaptive, efficient routing capability in WSNs. However, this would require further studies in the future work.
Appendix A

NS Trace Formats

<table>
<thead>
<tr>
<th>Event</th>
<th>Abbreviation</th>
<th>Flag</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wireless Event</td>
<td>s: Send</td>
<td>-t</td>
<td>double</td>
<td>Time</td>
</tr>
<tr>
<td></td>
<td>r: Receive</td>
<td>-Ni</td>
<td>int</td>
<td>Node ID</td>
</tr>
<tr>
<td></td>
<td>d: Drop</td>
<td>-Nx</td>
<td>double</td>
<td>Node X Coordinate</td>
</tr>
<tr>
<td></td>
<td>f: Forward</td>
<td>-Ny</td>
<td>double</td>
<td>Node Y Coordinate</td>
</tr>
<tr>
<td></td>
<td>c: Change</td>
<td>-Nz</td>
<td>double</td>
<td>Node Z Coordinate</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Ne</td>
<td>double</td>
<td>Node Energy Level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Nl</td>
<td>string</td>
<td>Network trace Level (AGT, RTR, MAC, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Nm</td>
<td>string</td>
<td>Drop Reason</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Ma</td>
<td>hexadecimal</td>
<td>Duration</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Ms</td>
<td>hexadecimal</td>
<td>Source Ethernet Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Md</td>
<td>hexadecimal</td>
<td>Destination Ethernet Address</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Mt</td>
<td>hexadecimal</td>
<td>Ethernet Type</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-P</td>
<td>string</td>
<td>Packet Type (arp, dsr, imep, tora, etc.)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-Pn</td>
<td>string</td>
<td>Packet Type (cbr, tcp)</td>
</tr>
</tbody>
</table>
Appendix B

Route Duration Statistics in Proactive Mobile Ad hoc Networks

The characteristics of MANETs, such as link duration and route duration, vary over time and be much more dynamic than for fixed networks. The transience of local and global network connectivity may significantly affect the performance of MANET routing protocols. So, assumptions that may be made about link duration and route duration in fixed networks may not hold for MANETs, and there may be causal effects observed in other metrics, such as loss and throughput.

Therefore, in order to provide a more robust service in MANETs, it is essential to understand the transient behaviour of MANET routing protocols and their impact on routing performance under various scenarios and factors, especially for proactive routing protocols such as OLSR.

This study investigates the route dynamics of Shortest-Path First (SPF) routing protocols in MANETs by examining the performance of OLSR. An analytical approach is developed to obtain the network statistics of link and route durations, including Probability Density Functions (PDFs) and Cumulative Density Functions (CDFs). We aim to gain insight into proactive MANET protocol performance under various mobility patterns, node velocity and node density.

Up till now, there has been no examination of route duration in proactive MANET routing protocols. Considering their difference in routing selection and maintenance, the route dynamics of proactive routing protocols may differ from those of reactive routing protocols. Proactive routing protocols consider optimal route selection, using algorithms such as Dijkstra’s Algorithm, so their behaviour is likely to be complex.

B.1 Experimental Methodology

This section explains the methodology used in this investigation, including performance metrics and the related measurement methods.

B.1.1 Performance Metrics

To evaluate the performance of a routing protocol, quantitative metrics need to be defined. In general, metrics such as link duration and link change interval are independent of the routing protocols. For example, link duration is associated with node property (i.e. radio range), node mobility (i.e. node
velocity, mobility patterns) and network properties (i.e. node density). Route duration, on the other hand, is determined by the route protocols, including route discovery and route selection algorithms. In the following paragraphs, we define two protocol-dependent metrics: route duration (RD) and route change (arrival) interval (RCI). The symbols used are as shown in Table B.1.

- **Route Duration (RD).** For a shortest route $r_{ij}$ between node $i$ and node $j$ with $k$ intermediate nodes, at time $t_1$, the route duration is the maximum period $[t_1, t_2]$ during which (a) each of the $k - 1$ links between the nodes is connected and symmetric; (b) the route represents the shortest path between the two nodes. Moreover, for each $\varepsilon > 0$, during the period $[t_1 - \varepsilon, t_2 + \varepsilon]$, either at least one of the $k - 1$ link does not exist, or there exists a new route $r'_{ij}$, such that $l'_{ij} < l_{ij}$.

- **Route Change Interval (RCI).** Any change in the set of routes may be due to the breakage of an established active route or due to re-selecting a new route. For nodes $i, j$, at time $t_1$, the duration of the route $r_{ij}$ is the maximum period $[t_1, t_2]$ during which there are no route breakage or route re-selection events. Moreover, for each $\varepsilon > 0$, during the period $[t_1 - \varepsilon, t_2 + \varepsilon]$, there exists at least one route change event.

### B.1.2 Approximation of Route Distribution

With a large set of samples collected, we use the relative frequency approach of probability theory to estimate the probability density functions (PDFs) of the RD and RCI across different mobility models. Once the PDFs are determined, we calculate the mean RD value and mean RCI value.

In order to approximate the distribution, this study uses standard curve fitting techniques to analyse the PDFs of the RD and RCI values across the different mobility models; the results of the curve fitting are listed in tables to facilitate further analysis.

This study evaluates goodness-of-fit by how well a statistical model fits a set of observations. Measures of goodness-of-fit typically summarise the discrepancy between observed values and the values expected under the model in question. Besides the visual examination of the fitted curve displayed, this study uses Sum of Squares due to Error (SSE), $R$-square, adjusted $R$-square ($aR$-square) and Root Mean Squared Error (RMSE) to assess the goodness-of-fit.

The $SSE$ statistic is the least squares error of the fit, with a value closer to zero indicating a better fit.

The $R$-square statistic is the square of the correlation between the response values and the predicted response value, with a value closer to 1 indicating that a greater proportion of variance is accounted for by the model.
B.2. Observations and Discussions

The \textit{adjusted R-square} statistic is generally the best indicator of the fit quality when additional coefficients are added to the models.

The \textit{RMSE} statistic measures the standard error of the fit, with a value closer to zero indicating a good fitting.

B.1.3 Simulation Set-up

The simulation environments used in this study are similar to those in Chapter 5. We use the OLSR implementation that runs in version 2.9 of NS2 with the ad-hoc networking extensions provided by CMU. The wireless channel type in this study is IEEE 802.11 wireless LAN with distributed coordination function (DCF). The channel has a capacity of 2Mb/s.

In particular, the radio radius is set varied across the set of values 50m, 100m, 150m, 200m, 250m, 300m, 400m in separate simulation runs.

A network of 20 nodes is used to simulate a low-density network, and a network of 50 nodes to simulate a high-density network. The nodes are placed in a $1000 \times 1000m^2$ field.

We use the Random Trip Mobility Model, Reference Point Group Model (RPGM), Freeway Model and Manhattan Model. The maximum velocity is set to 5m/s, 10m/s, 20m/s and 30m/s separately for each mobility model, while the minimum velocity is set to non-zero, e.g. 1m/s in our cases. The RPGM model has 4 groups, with velocity deviation ratio set to be 10\% of the maximum velocity. We use a range of velocity $v \leq 5m/s$ to simulate low-mobility networks (e.g. walking velocity or slow moving ground based vehicles), $v > 5m/s$ to simulate moderate and high mobility networks (e.g. ground-based vehicles or slow-moving airborne vehicles objects).

The mobility patterns of the Freeway Model, the RPGM and the Manhattan Model are generated by [103]. The mobility patterns of Random Trip Model are generated by [104].

B.2 Observations and Discussions

In this section, we present the observations on the route dynamics under various mobility models. Due to space restrictions, we focus on the observations of route duration distribution, with a radio range of 250m under the Random Trip model. The phenomena described below have been observed under all the other mobility models used in this work, including Freeway, Manhattan and RPGM models.

We have shown the values of \textit{SSE}, \textit{R-square}, \textit{aR-square}, and \textit{RMSE} to 3sf, so that we can show non-zero values for all our statistics. The calculated coefficients are at 95\% confidence.

B.2.1 Route Duration

From the distribution of route duration (Fig B.1, Fig B.2, Fig B.3, and Fig B.4)\textsuperscript{1} and the curve fitting results (Table B.2, Table B.3, Table B.4 and Table B.5) we can see that, under Random Trip model:

- When the velocity is relatively small (i.e. $v \leq 5m/s$), the RD distribution is non-exponential. For example, the $R - square$ is 0.051 when the maximum node velocity is set to be 1m/s, indicating a very small proportion of variance is accounted for by the exponential model.

\textsuperscript{1}In these figures, \textit{FR}(v = v_0) labels the curve of the fitting model for route duration distribution when the maximum node velocity is $v_0$. 

B.2. Observations and Discussions

Figure B.1: PDF of Route Duration for Random Trip Model (n = 20)

Figure B.2: PDF of Route Duration for Random Trip Model (n = 50)

Figure B.3: CDF of Route Duration for Random Trip Model (n = 20)
B.2. Observations and Discussions

![Figure B.4: CDF of Route Duration for Random Trip Model (n = 50)](image)

(a) Low Mobility
(b) High Mobility

Table B.2: Goodness of Fit (Fig B.1(a))

<table>
<thead>
<tr>
<th>(v)</th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.044</td>
<td>0.051</td>
<td>0.030</td>
<td>0.031</td>
</tr>
<tr>
<td>5</td>
<td>0.007</td>
<td>0.906</td>
<td>0.905</td>
<td>0.011</td>
</tr>
</tbody>
</table>

- With higher network mobility (i.e. \(v > 5 m/s\)), the RD could be approximated by an exponential distribution with 95% confidence and satisfactory goodness of fit (\(SSE < 0.003, R - square > 0.990, RMSE < 0.008\)).

In particular, as shown in Fig B.3(a) and Fig B.4(a), the route duration of a low-mobility network are relatively evenly distributed. The mean route duration in such networks is larger than that of high-mobility networks. With the increase of node velocity, the route duration decreases since mobility brings more frequent route changes, which leads to an exponential distribution.

By comparing Table B.2 with Table B.4, and comparing Table B.3 with Table B.5 we can see that, with the same node velocity, the route duration of high-density networks shows better exponential property than that of low-density networks. For example, when the node velocity is \(5 m/s\), the \(R - square\) of a low-density network (i.e. \(n = 20\)) is 0.906, while that of a high-density network (i.e. \(n = 50\)) is 0.992. This is because the link change rate increases with node density [52].

Note that the observations so far are based on the route duration sets with variable route lengths (\(L\)), the number of links/hops in the selected route. The route duration of specific route lengths is shown in Fig B.5, Table B.6 and Table B.7.

From the route duration of certain route lengths (Fig B.5) we can see that under the Random Trip Model:

- At high mobility, longer routes show better fit to an exponential distribution than shorter routes. As shown in Table B.7, the RD of short routes (i.e. \(L < 4\)) is not exponentially distributed, while the RD distribution of longer route (i.e. \(L \geq 4\)) can be approximated with an exponential distribution.
Table B.3: Goodness of Fit (Fig B.1(b))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 10$</td>
<td>0.002</td>
<td>0.986</td>
<td>0.986</td>
<td>0.006</td>
</tr>
<tr>
<td>$v = 20$</td>
<td>0.001</td>
<td>0.996</td>
<td>0.996</td>
<td>0.004</td>
</tr>
<tr>
<td>$v = 30$</td>
<td>0.001</td>
<td>0.997</td>
<td>0.996</td>
<td>0.005</td>
</tr>
</tbody>
</table>

Table B.4: Goodness of Fit (Fig B.2(a))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 1$</td>
<td>0.013</td>
<td>0.353</td>
<td>0.346</td>
<td>0.012</td>
</tr>
<tr>
<td>$v = 5$</td>
<td>0.001</td>
<td>0.992</td>
<td>0.992</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table B.5: Goodness of Fit (Fig B.2(b))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 10$</td>
<td>0.001</td>
<td>0.997</td>
<td>0.997</td>
<td>0.002</td>
</tr>
<tr>
<td>$v = 20$</td>
<td>0.001</td>
<td>0.997</td>
<td>0.997</td>
<td>0.003</td>
</tr>
<tr>
<td>$v = 30$</td>
<td>0.001</td>
<td>0.998</td>
<td>0.998</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table B.6: Goodness of Fit (Fig B.5(a))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 2$</td>
<td>0.013</td>
<td>0.182</td>
<td>0.137</td>
<td>0.027</td>
</tr>
<tr>
<td>$L = 3$</td>
<td>0.009</td>
<td>0.022</td>
<td>-0.005</td>
<td>0.016</td>
</tr>
<tr>
<td>$L = 4$</td>
<td>0.014</td>
<td>0.049</td>
<td>0.019</td>
<td>0.021</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>0.011</td>
<td>0.505</td>
<td>0.491</td>
<td>0.017</td>
</tr>
</tbody>
</table>

Table B.7: Goodness of Fit (Fig B.5(b))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$L = 2$</td>
<td>0.013</td>
<td>0.173</td>
<td>0.142</td>
<td>0.022</td>
</tr>
<tr>
<td>$L = 3$</td>
<td>0.008</td>
<td>0.466</td>
<td>0.450</td>
<td>0.015</td>
</tr>
<tr>
<td>$L = 4$</td>
<td>0.003</td>
<td>0.937</td>
<td>0.935</td>
<td>0.011</td>
</tr>
<tr>
<td>$L = 5$</td>
<td>0.003</td>
<td>0.972</td>
<td>0.970</td>
<td>0.012</td>
</tr>
</tbody>
</table>
B.2. Observations and Discussions

This is because longer routes involve more links and therefore have the route as a whole has a higher probability in suffering a broken link or a shorter (alternative) route being found at the next route recalculation.

- At low mobility, route duration with specific route lengths is non-exponential. For example, as shown in Table B.6, the $R - square$ value is smaller than 0.6 when the node velocity is 5 m/s, which indicates poor fit to the exponential model.

B.2.2 Route Change Interval

Similar phenomena have been observed in the distribution of route change interval (RCI).

For example, as shown in Fig B.6(b) and Table B.9, in a network with moderate or high mobility (i.e. $v \geq 5m/s$), the RCI can be approximated by an exponential distribution with 95% confidence and satisfactory goodness of fit ($SSE < 0.001, R - square > 0.990, RMSE < 0.004$). Meanwhile, the RCI in a low-mobility network can not (Fig B.6(a) and Table B.8).

Figure B.6: PDF of RCI for Random Trip Model ($n = 50$)
B.2. Observations and Discussions

Table B.8: Goodness of Fit (Fig B.6(a))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 1$</td>
<td>0.013</td>
<td>0.353</td>
<td>0.346</td>
<td>0.012</td>
</tr>
<tr>
<td>$v = 5$</td>
<td>0.001</td>
<td>0.993</td>
<td>0.993</td>
<td>0.003</td>
</tr>
</tbody>
</table>

Table B.9: Goodness of Fit (Fig B.6(b))

<table>
<thead>
<tr>
<th></th>
<th>SSE</th>
<th>R-square</th>
<th>aR-square</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v = 10$</td>
<td>0.001</td>
<td>0.997</td>
<td>0.997</td>
<td>0.002</td>
</tr>
<tr>
<td>$v = 20$</td>
<td>0.001</td>
<td>0.997</td>
<td>0.997</td>
<td>0.004</td>
</tr>
<tr>
<td>$v = 30$</td>
<td>0.001</td>
<td>0.998</td>
<td>0.998</td>
<td>0.003</td>
</tr>
</tbody>
</table>

B.2.3 Average Link Duration and Average Route Duration

The average link duration and the average route duration under Random Trip Model are shown in Fig B.7.

B.2.4 Comparison between SPF Routing and Reactive Routing

The route duration of SPF routing has similar distribution to that of reactive routing. However, the (mean) route duration of SPF routing is much smaller than that of reactive routing. The route duration of SPF routing shows better fit to an exponential than that of reactive routing. For example, as shown in [52], the average path duration of reactive routing for $v < 20m/s$ and $R = 250m$ is larger than 10s. However, the average route duration for SPF routing is less than 5s.

The other difference between these protocols is the route duration of SPF routing with specific route lengths show no fit to an exponential, while that of reactive routing does [52].
### Appendix C

## Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AODV</td>
<td>Ad hoc On-Demand Distance Vector Routing</td>
</tr>
<tr>
<td>ADV</td>
<td>Adaptive Distance Vector Routing</td>
</tr>
<tr>
<td>ANSN</td>
<td>Advertised Neighbour Sequence Number of OLSR</td>
</tr>
<tr>
<td>BHS</td>
<td>Broadcast based Handshake option of fast neighbour handshake algorithm</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Ratio</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CLT</td>
<td>Central Limit Theorem</td>
</tr>
<tr>
<td>CDF</td>
<td>Cumulative Distribution Function</td>
</tr>
<tr>
<td>CSMA</td>
<td>Carrier Sense Multiple Access</td>
</tr>
<tr>
<td>CSMA/CA</td>
<td>Carrier Sense Multiple Access with Collision Avoidance</td>
</tr>
<tr>
<td>DV</td>
<td>Distance Vector Routing</td>
</tr>
<tr>
<td>DSDV</td>
<td>Destination Sequenced Distance Vector Routing</td>
</tr>
<tr>
<td>DBF</td>
<td>Distributed Bellman-Ford algorithm</td>
</tr>
<tr>
<td>DSR</td>
<td>Dynamic Source Routing</td>
</tr>
<tr>
<td>DT_MIAD</td>
<td>Dynamic Timer Based on Multiplicative Increase Additive Decrease</td>
</tr>
<tr>
<td>DT_ODPU</td>
<td>Dynamic Timer Based on On-Demand Proactive Update</td>
</tr>
<tr>
<td>DCF</td>
<td>Distributed Coordination Function</td>
</tr>
<tr>
<td>FSR</td>
<td>Fisheye State Routing</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>kbps</td>
<td>kilo bits per second</td>
</tr>
<tr>
<td>LS</td>
<td>Link State Routing</td>
</tr>
<tr>
<td>LAN</td>
<td>Local Area Network</td>
</tr>
<tr>
<td>MANET</td>
<td>Mobile Ad hoc Network</td>
</tr>
<tr>
<td>MAC</td>
<td>Medium Access Control</td>
</tr>
<tr>
<td>MPR</td>
<td>Multi-Point Relay</td>
</tr>
<tr>
<td>NRO</td>
<td>Normalised Routing Overhead</td>
</tr>
<tr>
<td>NS-2</td>
<td>Network Simulator version 2</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>OLSR</td>
<td>Optimised Link State Routing</td>
</tr>
<tr>
<td>OSPF</td>
<td>Open Shortest Path First Routing</td>
</tr>
<tr>
<td>PDR</td>
<td>Packet Delivery Ratio</td>
</tr>
<tr>
<td>PDF</td>
<td>Probability Density Function</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RCI</td>
<td>Route Change Interval</td>
</tr>
<tr>
<td>RD</td>
<td>Route Duration</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource Reservation Setup Protocol</td>
</tr>
<tr>
<td>RWP</td>
<td>Random Waypoint Mobility Model</td>
</tr>
<tr>
<td>RTM</td>
<td>Random Trip Mobility Model</td>
</tr>
<tr>
<td>RRWP</td>
<td>Restricted Random Waypoint Model, part of RTM</td>
</tr>
<tr>
<td>RPGM</td>
<td>Reference Point Group Mobility Model</td>
</tr>
<tr>
<td>RMSE</td>
<td>Root Mean Squared Error</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request Query message of AODV</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply message of AODV</td>
</tr>
<tr>
<td>RERR</td>
<td>Route Error message of AODV</td>
</tr>
<tr>
<td>RTS/CTS</td>
<td>Request to Clear/Clear to Send</td>
</tr>
<tr>
<td>SSE</td>
<td>Sum of Squares due to Error</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SPF</td>
<td>Shortest Path First algorithm</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>TC</td>
<td>Topology Control</td>
</tr>
<tr>
<td>TBRPF</td>
<td>Topology Dissemination Based on Reverse-Path Forwarding</td>
</tr>
<tr>
<td>TORA</td>
<td>Temporally Ordered Routing Algorithm</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>UHS</td>
<td>Unicast based Handshake option</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>WRP</td>
<td>Wireless Routing Protocol</td>
</tr>
<tr>
<td>WLAN</td>
<td>Wireless Local Area Network</td>
</tr>
<tr>
<td>WiFi</td>
<td>IEEE 802.11 family</td>
</tr>
<tr>
<td>ZRP</td>
<td>Zone Routing Protocol</td>
</tr>
</tbody>
</table>
Bibliography


