LEABHARLANN CHOLÁISTE NA TRÍONÓIDE, BAILE ÁTHA CLIATH | TRINITY COLLEGE LIBRARY DUBLIN
Ollscoil Átha Cliath | The University of Dublin

Terms and Conditions of Use of Digitised Theses from Trinity College Library Dublin

Copyright statement

All material supplied by Trinity College Library is protected by copyright (under the Copyright and Related Rights Act, 2000 as amended) and other relevant Intellectual Property Rights. By accessing and using a Digitised Thesis from Trinity College Library you acknowledge that all Intellectual Property Rights in any Works supplied are the sole and exclusive property of the copyright and/or other IPR holder. Specific copyright holders may not be explicitly identified. Use of materials from other sources within a thesis should not be construed as a claim over them.

A non-exclusive, non-transferable licence is hereby granted to those using or reproducing, in whole or in part, the material for valid purposes, providing the copyright owners are acknowledged using the normal conventions. Where specific permission to use material is required, this is identified and such permission must be sought from the copyright holder or agency cited.

Liability statement

By using a Digitised Thesis, I accept that Trinity College Dublin bears no legal responsibility for the accuracy, legality or comprehensiveness of materials contained within the thesis, and that Trinity College Dublin accepts no liability for indirect, consequential, or incidental, damages or losses arising from use of the thesis for whatever reason. Information located in a thesis may be subject to specific use constraints, details of which may not be explicitly described. It is the responsibility of potential and actual users to be aware of such constraints and to abide by them. By making use of material from a digitised thesis, you accept these copyright and disclaimer provisions. Where it is brought to the attention of Trinity College Library that there may be a breach of copyright or other restraint, it is the policy to withdraw or take down access to a thesis while the issue is being resolved.

Access Agreement

By using a Digitised Thesis from Trinity College Library you are bound by the following Terms & Conditions. Please read them carefully.

I have read and I understand the following statement: All material supplied via a Digitised Thesis from Trinity College Library is protected by copyright and other intellectual property rights, and duplication or sale of all or part of any of a thesis is not permitted, except that material may be duplicated by you for your research use or for educational purposes in electronic or print form providing the copyright owners are acknowledged using the normal conventions. You must obtain permission for any other use. Electronic or print copies may not be offered, whether for sale or otherwise to anyone. This copy has been supplied on the understanding that it is copyright material and that no quotation from the thesis may be published without proper acknowledgement.

FLEXIBILITY IN AD HOC NETWORKS

Timothy Kirby Forde

A thesis submitted to the University of Dublin
for the degree of Doctor of Philosophy

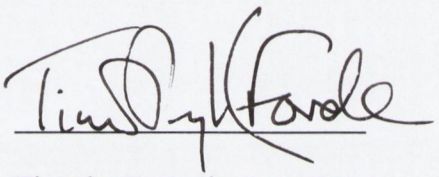
Department of Electronic and Electrical Engineering,
Trinity College, University of Dublin

2005

TRINITY COLLEGE
13 NOV 2006
LIBRARY DUBLIN
THESIS
B067

I declare that the work described in this thesis is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

I agree that Trinity College Library may lend or copy this thesis upon request.

Signed: 

Timothy K. Forde, B.A., B.A.I.

July 2005

Ar son mo thuismitheorí 's mo shean thuismitheorí

Summary

This thesis represents a step forward in our understanding of what flexibility, adaptability and reconfigurability means at the network-layer of mobile, wireless, ad hoc networks. Mobile, wireless, ad hoc networks are collections of autonomous routers that have the ability to dynamically and rapidly form networks without the use of any centralised network infrastructure using wireless communication technologies. Ad hoc routing protocols are at the core of enabling ad hoc networks as they allow remote nodes to communicate without resorting to primitive flooding techniques. Ad hoc networks, by their nature, will experience divergent user scenarios and networking conditions that demand routing protocols which are tailored to their requirements. However, to date, the static configuration of one routing protocol across the entire network has been the norm. This thesis argues that such a rigid approach constrains the temporally and spatially divergent nature of the system. This thesis shows that a more flexible approach to the design of the network-layer for mobile wireless ad hoc networks is both necessary and feasible. The thesis explores the need for such flexibility in ad hoc networks and presents an enabling system in the form of a novel network-layer which incorporates multiple routing protocols.

The design of this flexible multi-protocol network-layer demonstrates three important facts. Firstly, it establishes that it is feasible to design a flexible system at the network-layer of an open, mobile, wireless, ad hoc network that enables autonomous nodes to configure themselves to use a routing protocol, chosen from a suite of available routing protocols, which is suited to the prevailing network conditions that they observe. Secondly, it is shows that it is feasible to design such a system in a manner that preserves the inherent optimised characteristics of the underlying ad hoc protocols by using signalling techniques which are compatible with ad hoc routing protocols in use. Thirdly, it demonstrates that it is feasible to design a flexible system that allows a network to configure itself on a regional basis when networking conditions are not globally uniform and when one protocol is not optimal for the entire network such that regional variation in the configuration of the network does not prevent nodes situated in heterogeneously configured regions from communicating on an end-to-end basis in a multi-hop peer-to-peer manner.

The multi-protocol network-layer has components that facilitate reactive node auto-configuration, proactive node reconfiguration and interdomain ad hoc routing. It is designed for a real ad hoc network system, the Dublin Ad hoc Wireless Network (DAWN). The auto-configuration protocol uses a combination of stateless auto-configuration techniques, currently used in the area of address auto-configuration, and localised self-stabilisation techniques to maintain a stable and legal global state with regard to the configuration of the nodes' network-layers. Essentially, this protocol provides a lightweight, *just-in-time* correction facility for autonomous nodes which guarantees the correction of their configuration immediately upon detecting a fault and just in time to allow the node to provide ad hoc routing services. This protocol addresses the difficulties of coordinating a multi-state, asynchronous, message-passing, distributed system such as that characterised by a wireless, mobile ad hoc network and provides a stable platform on which to build the proactive reconfiguration protocol. This novel auto-configuration protocol incorporates a number of techniques that may be applied to other reconfigurable services, protocols and applications in similarly challenging settings.

The node reconfiguration protocol presents a more flexible approach to distributed consensus with a modified version of an innovation-decision process to provide a solution for network-wide conflict-free node

reconfiguration. While the distributed consensus literature delineates rigid bounds on the degree to which distributed coordination is possible in an ad hoc networking system, this thesis presents a novel framework that builds on the social-science theory of diffusion of innovations, a theory which describes a flexible decision-making model. The reconfiguration protocol enables nodes to autonomously and collaboratively choose their network-layer routing protocol having regard to the prevailing network conditions.

The temporal and spatial diversity of the network-layer may lead to the creation of an ad hoc internetwork of autonomous domains: each domain using an interior routing protocol which is optimally suited to the prevailing demands. Maintaining multi-hop, peer-to-peer connectivity in such a heterogeneous internetwork of autonomous and dynamically configured systems is a non-trivial task. This thesis presents an extensible framework that builds on the existing proven and robust interior ad hoc routing protocols, such as AODV, DSR and OLSR, extending their design philosophies to the task of seamless ad hoc interdomain routing. The framework's design takes a *completely* distributed approach, in keeping with the robust design philosophy of the underlying interior routing protocols. The framework makes use of a *broker* that sits in boundary nodes between different ad hoc network domains. This broker is responsible for ensuring that the scalability of the internetwork is maintained by throttling the rate at which control packets pass from one network into another. The broker is also responsible for transforming packets as they pass from one region into another so that they may be processed in the receiving region.

Acknowledgements

Firstly, I thank Linda Doyle for giving me the opportunity to explore this area of research and to produce this thesis. I also thank her for her boundless enthusiasm, encouragement, optimism, vision and her *almost* limitless patience that helped me to complete this work. Linda is a *great* person to work with and made my time in College very enjoyable.

I also thank Prof. Donal O'Mahony and the other members of the Network and Telecommunications Research Group for the collaborative discussions, meetings and projects that have allowed me to develop this work.

I also thank the guys in the lab; Brian, Declan, Keith, Ken, Philip,Yan Ming for making it a fun place to work and for providing a sounding-board for many of my ideas. Thanks to Philip, for both encouraging me to start the thesis and for helping out along the way.

I thank other Department staff and colleagues for making Arás an Phiarsaigh an enjoyable place to work; in particular I thank Anil, Conor, Nora, Robbie and *others*, for their good humour and ability not to talk shop.

I also thank my friends and family for showing absolutely no interest in mobile ad hoc networking for the last few years. It was a welcome relief to have the company of people with absolutely no interest in issues pertaining to multi-hop peer-to-peer routing. Thanks to David and Sarah for proofing the thesis.

Finally, I owe a great deal of gratitude to my parents, Liam and Martina, for providing me with all kinds of *very generous* support throughout my time here in Trinity. *Enjoy the read.*

Contents

1 Introduction 1

1.1 Introduction 1
1.2 Ad Hoc Networking in Dynamic, Diverse and Open Environments 2
1.3 Multiple Routing Protocols, Multiple Networking Environments 3
1.4 Flexibility in Ad Hoc Networks 4
1.5 The Thesis 6
1.6 Contributions 6
1.7 Thesis Overview 7

2 Ad Hoc Networks 3

2.1 Introduction 3
2.2 An Overview of Mobile Wireless Ad Hoc Networks 3
2.2.1 Distinguishing Traits and Challenging Features of Ad Hoc Networks 5
2.2.2 Applications of Ad Hoc Networks 7
2.2.3 Enabling Technologies for Ad Hoc Networks 8
2.3 Multi-Hop Ad Hoc Routing Protocols 12
2.3.1 Proactive Protocols 14
2.3.2 Reactive Protocols 15
2.3.3 Alternative Hybrid and Location-based Protocols 15
2.4 Evaluation of Multi-Hop Ad Hoc Network Routing Protocols 16
2.4.1 An Overview of Network Simulation and Evaluation Techniques 17
2.4.2 Variable Evaluation Parameters 19
2.4.3 Performance Metrics 24
2.4.4 A Survey of Published Evaluations 25
2.4.5 Summary of Published Evaluations 38
2.5 Ad Hoc Networks for the Future 48
2.6 Ad Hoc Networks in an Urban Context 50
2.7 Summary 55

3 Requirements of a Flexible Multi-Protocol Network-Layer 57

3.1 Introduction 57
3.2 Flexible Multi-Protocol Network-Layers 58
3.2.1 Flexibility to Enable Auto-Configuration: Node Initialisation and Node Join 60
3.2.2 Flexibility to Enable Auto-Configuration: Node Migration and Standalone Network Mergers 63

3.2.3 Flexibility to Enable the Ad Hoc Internetworking of Heterogeneous Standalone Ad Hoc Networks 65
3.2.4 Flexibility to Enable the Reconfiguration of Networks 67
3.3 Summary 69

4 Towards a Flexible Multi-Protocol Network-Layer 71

4.1 Introduction 71
4.2 Network-Layer Design 71
4.3 Mobile Ad Hoc Network-Layer Design 75
4.4 Future Mobile Ad Hoc Network-Layer Design 77
4.5 Design Restrictions 79
4.5.1 Assumptions 79
4.5.2 Propositions 81
4.6 The Basic Network-Layer 83
4.6.1 Routing Protocols 83
4.6.2 Route Cache 84
4.6.3 Neighbourhood Table 85
4.6.4 Encapsulating Network-Layer Packet 86
4.7 Design Platform 88
4.8 Summary 90

5 Enabling Observation & Analysis 91

5.1 Introduction 91
5.2 Observing the Networking Environment 92
5.2.1 Observable Events 95
5.2.2 Passive, Myopic Observation 97
5.3 Indicative Trials 99
5.4 Performance Profile Comparison 101
5.5 Myopic Measurements 104
5.6 Summary 110

6 Enabling Auto-Configuration 111

6.1 Introduction 111
6.2 Auto-Configuration: Initialisation, Join, Migration & Merging 112
6.2.1 Objectives 112
6.2.2 Challenges 112
6.2.3 Design Blueprint 114
6.3 A Node's Perspective through the Lens of Ambient Signalling: Network Configuration 114
6.4 Techniques of Interest 114

6.4.1 Ad Hoc Network Management 114
6.4.2 Address and Router Auto-Configuration 114
6.4.3 Self-Stabilisation 114
6.4.4 Summary 114
6.5 The Auto-Configuration Component 114
6.5.1 Overview of Component 114
6.5.2 Local State Variables and Node Status Messages 114
6.5.3 Stateless Configuration Initialisation (SCI) Protocol, Region Sequence Number .. 114
6.5.4 Configuration Conflict Detection and Resolution (CCDR) Protocol 114
6.6 Summary 114

7 Enabling Reconfiguration 114

7.1 Introduction 114
7.2 Reconfiguration 114
7.2.1 Objectives 114
7.2.2 Challenges 114
7.2.3 Design Blueprint 114
7.3 Distributed Consensus 114
7.4 Diffusion of Innovations: From Innovators to Laggards, Knowledge to Implementation 114
7.5 The Reconfiguration Component 114
7.5.1 Overview of the Component's Protocol Design 114
7.5.2 Regional Reconfiguration and the Auto-Configuration Component 114
7.5.3 Opportunistic Communication 114
7.6 Reconfiguration-Decision Protocol (RDP) 114
7.6.1 The Persuasion Stage 114
7.6.2 Soft-State Decision Making 114
7.6.3 The Decision and Implementation Stages 114
7.6.4 Creep: Trial-By-Others 114
7.7 Demonstrative Experiments 114
7.7.1 Changing Mobility Conditions 114
7.7.2 Changing Node Degree Conditions 114
7.7.3 Changing Traffic Conditions 114
7.8 Summary 114

8 Enabling Ad Hoc Internetworking 114

8.1 Introduction 114
8.2 Ad Hoc Internetworking 114
8.2.1 Objective 114
8.2.2 Challenges 114
8.3 Inter-Regional Multi-Protocol Routing 114

8.3.1 Home Networks, External Networks, Boundary Nodes and External Routes 114
8.3.2 Brokering Inter-Regional Route Discovery, Packet-Switching and Route Maintenance: An Overview 114
8.3.3 Controlling Diffusion into the External Network 114
8.3.4 Infusing External Routing Information into the Home Network 114
8.4 Inter-Regional Route Discovery 114
8.4.1 Route Discovery: Scenario One 114
8.4.2 Route Discovery: Scenario Two 114
8.5 Inter-Regional Packet Forwarding and Route Maintenance 114
8.5.1 Packet Switching and Route Maintenance: Scenario One 114
8.5.2 Packet Switching and Route Maintenance: Scenario Two 114
8.6 Summary 114

9 Conclusions 114

9.1 Introduction 114
9.2 Review of Contributions 114
9.3 Future Work 114
9.3.1 Disaggregated Network Architectures 114
9.3.2 Volatile Systems 114
9.3.3 Service-Driven Protocol Stacks 114
9.3.4 Cost-Benefit Comparison of Competing Network Solutions 114
9.4 Conclusion 114

10 Bibliography 114

A Core Ad Hoc Routing Protocols 114

B Self-Stabilisation by Local Checking and Local Correction 114

Figures

Figure 2-1 Nodes May Join or Leave an Ad Hoc Network at Any Time and at Will 4

Figure 2-2 An Ad Hoc Network May Become Partitioned Due to the Mobility of its Nodes 4

Figure 2-3 Two Networks of Nodes with Small Transmission Radii (left) and Larger Radii (right) 20

Figure 2-4 Determining the Maximum Diameter (in Hops) of a Network 21

Figure 2-5 Sample Mobility Models: Random Waypoint (left), Nomadic Community (right) 22

Figure 2-6 Sample Node Speeds 23

Figure 1-7 A Pedestrianised Shopping Street 53

Figure 1-8 An Urban Park 54

Figure 3-1 Illustration of a Park and an Adjacent Street: Morning 58

Figure 3-2 Illustration of a Park and an Adjacent Street: Afternoon 59

Figure 3-3 Illustration of a Park and an Adjacent Street: Evening 59

Figure 3-4 A single node boots up and auto-configures itself 61

Figure 3-5 A group of nodes boot up and auto-configure themselves 61

Figure 3-6 Node Migration between Two Standalone Networks 63

Figure 3-7 Complete Merger of Two Standalone Networks 64

Figure 3-8 Partial Merger of Two Standalone Networks 66

Figure 3-9 Internetworking of Two Heterogeneous Standalone Networks 67

Figure 3-10 A Network of Statically Configured Single-Protocol Nodes 68

Figure 3-11 A Network Composed of Heterogeneously Configured Regions 69

Figure 4-1 TCP/IP Internet layer 74

Figure 4-2 Typically Suggested Structure of an Ad Hoc Router Node 76

Figure 4-3 Three Examples of Ad Hoc Routers Suggested in the Literature; AODV Router (a), DSR Router (b) and OLSR Router (c) 77

Figure 4-4 A Basic Blueprint to Guide the Flexible Component Design 79

Figure 4-5 Three Distinct Route Cache Structures and Associated Neighbourhood Table Entries 85

Figure 4-6 Network-Layer Packet Header 87

Figure 4.7 Design Platform Communication Stack Organisation 89

Figure 5-1 Blueprint for Design of Flexibility Components: Network Observation 92

Figure 5.2 Sample of Global Performance Results for AODV, DSR and OLSR 103

Figure 5.3 AODV Node Degree, Link Changes and Link Duration 105

Figure 5.4 DSR Node Degree, Link Changes and Link Duration 106

Figure 5.5 OLSR Node Degree, Link Changes and Link Duration 107

Figure 5.6 Extended Evaluations: 1200s and 18000s 109

Figure 6-1 A Single Node Boots-up and Auto-Configures Itself by Joining the Network 113

Figure 6-2 Node Migration between Two Standalone Networks 113

Figure 6-3 Two Standalone Networks Merge 114

Figure 6-4 Blueprint for Design of Auto-Configuration Component 114

Figure 6-5 Link Asymmetry may Occur in the Incoming or Outgoing Direction 114

Figure 6-6 Non-Mutuality of Neighbourliness 114

Figure 6-7 Illustration of Isolated and Connected DSR and OLSR Nodes 114

Figure 6-8 Node Isolation and Signalling Activity: The Impact on Reaction to Network Events 114

Figure 6-9 System Viewed as Union of Overlapping Subsystems 114

Figure 6-10 Varghese's Local Checking and Correction Example 114

Figure 6-11 Nodes Can Change Protocols Dynamically and Operate Two Protocols Simultaneously 114

Figure 6-12 A Network, Two Regions and A Link Subsystem 114

Figure 6-13 Local State of the (u, v) Link Subsystem 114

Figure 6-14 Node Status Message Extension 114

Figure 6-15 Network Identifiers Must be Partition-Safe 114

Figure 6-16 Unique Protocol-to-Protocol Transition Identifiers 114

Figure 6-17 The Three Possible Region Sequence Number Cycles 114

Figure 6-18 Two-Round Initial Random Protocol Selection 114

Figure 6-19 Region Sequence Numbers avoid Corrective Thrashing 114

Figure 6-20 One-Way Checking: Interpretation of Node Status Messages 114

Figure 6-21 Conflict Detection and Resolution Protocol Pseudo-Code, Predicate One: Part I of II 114

Figure 6-22 *Conflict Detection and Resolution Protocol Pseudo-Code, Predicate One: Part II of II*..... 114

Figure 6-23 *Conflict Detection and Resolution Protocol Pseudo-Code, Predicate Two*..... 114

Figure 6-24 *CCDR Examples - (a), (b) and (c)*..... 114

Figure 6-25 *CCDR Examples - (d), (e) and (f)*..... 114

Figure 6-26 *CCDR Examples - (g), (h) and (i)*..... 114

Figure 6-27 *CCDR: Node Migration and Join*..... 114

Figure 6-28 *CCDR: Network Mergers*..... 114

Figure 7-1 *Blueprint for Design of Reconfiguration Component*..... 114

Figure 7.2 *Three Decision Models*.....174

Figure 7.3 *Regional Fragmentation*.....176

Figure 7.4 *Early Adopters Have Similar Experiences*..... 179

Figure 7.5 *Urban Networking Scenario*.....181

Figure 7.6 *Reconfiguration Status Header*..... 182

Figure 7.7 *Neighbourhood Table: ESV Data and Soft-State Decision*..... 182

Figure 7.8 *Decision Hierarchy*.....183

Figure 7-9 *Early Adopter Decision: Node's and Neighbours' Observations Only*..... 184

Figure 7-10 *Early Majority Decision: Early Adopter Neighbours Only*..... 186

Figure 7-11 *Trial-By-Others: Creep*..... 188

Figure 7-12 *Reconfiguration in Response to Changing Node Mobility*..... 190

Figure 7-13 *Reconfiguration in Response to Changing Node Degree*..... 192

Figure 7-14 *Reconfiguration in Response to Changing Offered Traffic Loading*.....193

Figure 8-1 *Hierarchical Internet Structure*..... 114

Figure 8-2 *Internet to Ad Hoc Network Connections through Fixed Gateways*..... 114

Figure 8-3 *Multiple Heterogeneous Ad Hoc Networks Require Dynamic Interconnection*..... 114

Figure 8-4 *Home Networks, External Networks, Boundary Nodes and External Routes*..... 114

Figure 8-5 *End-to-End Route Discovery Process: Phase One*..... 205

Figure 8-6 *External Request Table*.....206

Figure 8-7 *External Route Cache Structure*..... 210

Figure 8-8 *End-to-End Route Discovery Process: Phase Two*..... 211

Figure 8-9 AODV_Route_Reply_External_Network Message Format..... 212

Figure 8-10 Modified OLSR HNA Message Format 212

Figure 8-12 Scenario One: Network Diagram and Route Discovery Messages 217

Figure 8-13 Scenario Two: Network Diagram and Route Discovery Messages 223

Figure 8-14 Scenario One: Network Diagram and Packet-Switching, Route Maintenance Messages 227

Figure 8-15 Scenario One: Network Diagram and Packet-Switching, Route Maintenance Messages 232

Figure 9.1 Control Plane Disaggregation over Space 239

Figure 9.2 Control Plane Disaggregation over Time.....240

Figure A-1 *AODV Protocol State Data, Tables, Route Cache and Data Buffers* *114*

Figure A-2 *AODV Control Messages* *114*

Figure A-3 *AODV Networking Example* *114*

Figure A-4 *DSR Protocol State Tables, State Data, Route Cache and Data Buffers* *114*

Figure A-5 *DSR Control Messages* *114*

Figure A-6 *DSR Networking Example* *114*

Figure A-7 *OLSR Protocol State Tables, State Data, and Route Cache* *114*

Figure A-8 *OLSR Control Messages* *275*

Figure A-9 *OLSR Networking Example* *114*

Figure B-1 *One Global Network, Eight Local Link Subsystems* *280*

Figure B-2 *A Network Featuring An Illegal Link Subsystem* *281*

Figure B-3 *Two Overlapping Link Subsystems* *281*

Figure B-4 *Two-Way Snapshot and One-Way Checking* *282*

Tables

Table 2-1 *Commercial Wireless Transmission Options* *12*

Table 2-2 *Selective Summary of Ad Hoc Routing Protocol Features* *16*

Table 5-1 *RWP Mobility Model Parameter Values* *100*

Table 5-2 *Simulation Area, Node Density and Node Degree Parameter Values* *100*

Table 5-3 *Traffic Model Parameter Values* *101*

1 Introduction

1.1 Introduction

This thesis shows that a more flexible approach to the design of the network-layer for mobile wireless ad hoc networks is both necessary and feasible. The thesis explores the need for such flexibility in ad hoc networks and presents an enabling system in the form of a flexible multi-protocol network-layer. Throughout this thesis the terms *mobile wireless ad hoc network* and *ad hoc network* are used interchangeably; however, the thesis focuses on ad hoc networks that are mobile and which use wireless transmission media as these ad hoc networks provide the greatest degree of networking freedom and also pose the greatest challenges for the development of new protocols, services and applications.

A mobile wireless ad hoc network is an autonomous collection of routers that have the ability to dynamically and rapidly form networks without the use of any centralised network infrastructure using wireless communication technologies. Mobile wireless ad hoc networking technologies originated in the 1970s in the U.S. military context. However, it was the recent advent of powerful, mobile, commercially available computing platforms that led to a resurgence of interest in this area. As the technology moves from closed and controlled military environments to open civilian scenarios that present many opportunities for applications and services, it brings with it an array of new challenges. While ad hoc networking solutions are robust and suitable for many scenarios that current planned and centralised technologies are not designed for, the autonomous nature of open ad hoc networks poses many new challenges and requires a new approach to the design of ad hoc networking systems.

This thesis addresses some of these challenges by proposing a flexible ad hoc networking system. Notably, it addresses issues that arise at the network-layer in choosing an appropriate ad hoc multi-hop routing protocol.

The conditions in which the network operates have a significant impact on the choice of routing protocol. This thesis motivates a more flexible approach to the design of the network-layer for mobile wireless ad hoc networks, such that the network can adapt itself to the prevailing networking conditions which may exist in the diverse array of possible networking scenarios. Such flexibility will enable ad hoc networks to take advantage of a number of distinct ad hoc routing protocols which are suited to divergent networking conditions. Ad hoc networks should not only be able to make ad hoc choices with regard to the multi-hop routes that they discover and use, but they should also be enabled to make ad hoc choices about the routing protocols that they employ. This is possible within the context of a system built upon a flexible multi-protocol network-layer.

1.2 Ad Hoc Networking in Dynamic, Diverse and Open Environments

Mobile wireless ad hoc networks have many projected and potential uses. They range from tactical battlefield networks and disaster-relief networks to community networks and interactive artistic installations. There are many opinions on the place of ad hoc networks within the broader communications field. Some see ad hoc networking as a disruptive technology that threatens the order and discipline of current hierarchical services such as the Internet or GSM/UMTS services. Others see ad hoc networking as a technology that extends the range and reach of such technologies into spaces and scenarios that these centralised technologies could not penetrate.

From an engineering point-of-view, ad hoc networking may be viewed as an evolutionary technology that will lead existing networking conventions into the mobile wireless domain, leveraging the existing networking standards by tweaking them for a wireless and mobile environment. On the other hand, mobile wireless ad hoc networks, and the service and application opportunities that they provide, may also be viewed as the chance to make a step-change in the way in which networking technologies are designed and operated. This thesis takes the later view.

There are many challenging issues within the research field of ad hoc networking. Ad hoc networks are inherently distributed systems; the conventions associated with conventional stable centralised networks are turned on their head. Many networking issues have to be addressed anew. Innovative solutions for security, transport and addressing issues have already been developed to counter the difficulties of this challenging domain. However, the network-layer solutions are possibly the most crucial components of the ad hoc network as the network-layer binds the disparate elements of the network together. It is the service provided by multi-hop routing protocols that allows remote nodes in an ad hoc network to communicate on a multi-hop peer-to-peer basis. The *vast majority* of higher-level applications rely on the successful operation of the routing protocol to enable their services.

It is the *ad hoc* quality of ad hoc networks that uniquely distinguishes them from other networking standards and contexts. Ad hoc networks are built for ad hoc purposes; they are inherently transient in nature, addressing a temporary or unforeseen networking demand. The need for ad hoc networks may arise at anytime in any place, hence the ability of the constituent nodes to form networks in an ad hoc manner. Other permanent, planned, and generally, centrally administered networks address needs that may be foreseen or

which are long-lived. Ad hoc networks are further characterised by the autonomy of the constituent nodes, with regard to their movement and whether they are powered-on or off. Autonomous ad hoc nodes can move anywhere at anytime without reference to centralised authorities, allowing networks to be formed with random topologies which do not conform to the conventional planned, and possibly optimal, network topologies.

As ad hoc networks are formed by collaborating autonomous nodes, realistically, an ad hoc network in a civilian environment will be open, allowing nodes to leave or join the network at will. This is in contrast to the oft-cited military or rescue-focused approaches of ad hoc networks, where networks are closed for security purposes. In such closed environments, the movement of nodes and the services provided may be marshalled according to need and according to strict hierarchical order which reflects the social order of the controlling structure. Imposing such order in an open non-hierarchical civilian environment may be anathema to the participants in such a network.

Regardless, discussion about the future of ad hoc networks remains conjectural; there is much ambiguity as to the form that ad hoc networks will take or on what kind of devices the ad hoc nodes will manifest themselves. With such uncertainty and possible variety at issue, ad hoc networking systems must be flexible enough to cope with dynamic, open and diverse operating environments which cannot be presently conceived. Such flexibility must inevitably manifest itself in the network-layer technologies of such systems as it is the network-layer that forms the bed-rock underlying all mobile ad hoc network services and applications.

1.3 Multiple Routing Protocols, Multiple Networking Environments

The network-layer is one of the most important layers of the communication stack in any networking context. Within the context of mobile ad hoc networking it is the routing protocol component of the network-layer which is *the* most important part of the network-layer. The main networking challenges posed by a mobile, wireless, ad hoc network centre around the need to use efficient signalling techniques to enable the routing protocols to converge rapidly in response to the dynamic network topology. Signalling in a contentious wireless environment should be judicious and efficient due to the innate power and bandwidth constraints of such networks. Regardless of the specific mechanisms employed by the routing protocol, it is this component that enables nodes to initially discover the changing topology of the network and to maintain or rediscover the topology in an efficient manner as the nodes continue to move about in an unplanned manner. From the point-of-view taken in this thesis, the routing protocol is the primary network-layer component; without a functioning routing protocol there is no functioning network.

It is shown that the network conditions that greatly affect the performance of ad hoc routing protocols are the relative mobility of the nodes, the degree of connectivity of the nodes and the traffic that is loaded onto the network. These factors are independent of the routing protocols; they are characteristics of the networking environment beyond the control of the network-layer. The literature clearly shows that none of the available protocols enable the adoption of a one-size-fits-all approach within the context of a networking environment that exhibits such spatial and temporal diversity of network conditions as an open, autonomous, mobile, wireless, ad hoc network.

While many multi-hop ad hoc routing protocols have been proposed and evaluated in recent years, individual protocols have been shown to be suited to different combinations of networking conditions. The protocols differ in the techniques they use to discover, maintain and use multi-hop routing information. Some use distributed distance-vector techniques and optimised link-state techniques while others use source-routing techniques. These different route discovery and route maintenance approaches have the effect of enabling some protocols do deal effectively with some combinations of networking conditions, but to struggle in others. While it may be possible to adaptively tune certain local parameters of some routing protocols such that the effectiveness of the protocol is improved to a degree, such local tuning does not change the fundamental routing techniques underlying the protocol. When there are significant changes in the networking conditions, such local tuning devices will not suffice. Furthermore, local tuning techniques may lead to the establishment of greedy nodes which unilaterally increase the control overhead they incur on the network to their own ends, thereby negatively affecting the performance of other nodes.

So, in the absence of an ability to effect control over the dominant networking conditions, the *network* should instead configure itself in such a way that it can minimise the negative effects of these conditions. To this end, this thesis presents an argument that an ad hoc network should be enabled to make ad hoc decisions about the choice of routing protocol at the network-layer, such that it may take advantage of the array of routing protocols that are being developed. This thesis presents a flexible system designed around a network-layer which has access to a suite of routing protocols at runtime. The routing protocols address different networking conditions by utilising alternative techniques to discover and maintain multi-hop routes. The aim of the system is to enable the network to configure the network-layer to use the routing protocol most appropriate to the underlying networking conditions as described by the network's node mobility, node degree and traffic loading characteristics. In this system, three maturing distributed routing protocols are used to elucidate the issues surrounding the development of a flexible multi-protocol network-layer. The protocols are the reactive distance-vector-based Ad Hoc On-Demand Distance-Vector (AODV) protocol, the reactive source-routing-based Dynamic Source Routing (DSR) protocol and the proactive link-state-based Optimised Link-State Routing (OLSR) protocol. The AODV, DSR and OLSR protocols have been used for this thesis in order to tease out and develop a multi-protocol network-layer that has routing components suited to a wide range of networking conditions. The operation of each protocol is sufficiently different to warrant that a clear description of each protocol is provided in Appendix A for those who are unfamiliar with the three protocols. As the protocols use different routing mechanisms, they demand that the techniques used in the system are as generic and widely applicable as possible, addressing the needs of source-routed reactive protocols (DSR), distributed reactive protocols (AODV) and time-triggered proactive protocols (OLSR). Other ad hoc routing protocols could be used, in this work, in place of the protocols chosen. Indeed, a location-based protocol, i.e. one using a GPS-like system, could also have been used if the weakening of the design criteria discussed later in this thesis was acceptable.

1.4 Flexibility in Ad Hoc Networks

Enabling the network to make dynamic choices about the type of routing protocol to use at any given time demands that the structure of the network-layer is redesigned. A dynamically reconfigurable network will revolve around a multi-protocol network-layer which allows the network to fragment into heterogeneously

configured regions as and when networking conditions dictate. The spatial and temporal diversity of networking conditions that may be experienced throughout a network mean different protocols may be suitable in different regions of the network at different times.

Enabling such dynamic reconfigurability throws up a number of complex issues which have not had to be addressed in designing network-layer structures heretofore. A network built upon the bed-rock of a flexible multi-protocol network-layer must continually manage itself so that its choice of routing protocol reflects the underlying networking conditions and so that each node connected to the network is configured such that it may communicate with every other node, subject only to normal networking constraints, on a multi-hop peer-to-peer basis. The configuration of the ad hoc network is no longer determined by the hit-and-miss static configuration of each typical node's single protocol prior to its deployment.

Creating a system that enables nodes to choose their own ad hoc routing protocol instead of statically pre-configuring the ad hoc routing protocol demands that the network-layer is enabled to manage itself throughout the life-cycle of each node. In advance of experience of the networking conditions in an ad hoc network, multi-state nodes may configure themselves at their own discretion. However, in order to enable nodes to communicate on a multi-hop peer-to-peer basis, nodes must use the same routing protocol. Therefore, the flexible multi-protocol network-layer must enable the nodes to organise their choice of routing protocol when faced with naturally-occurring disruptive network events such as nodes booting-up in an established network, nodes joining a network with a different network-layer configuration, nodes migrating from one network region to another, and the merging of networks that have chosen different ad hoc routing protocols. Of course, in addition to dealing with these events and the possible conflicts between nodes' network-layer configurations, nodes must also be enabled to reconfigure themselves in response to changes in the networking conditions.

Furthermore, as the network's nodes are enabled to configure themselves in groups on a regional basis in response to the prevailing networking conditions, they will partition the network by virtue of their network-layer heterogeneity. If two groups of nodes, using different ad hoc routing protocols, come into transmission range of each other, they cannot share multi-hop routing information as the protocols they use can only be used to exchange routing information with like-configured nodes. In conventional wired networks, heterogeneous autonomous systems are enabled to share routing information and data by means of internetworking technologies.

It is evident that a network-layer which offers such choice to the nodes requires a degree of sophistication not previously found in network-layers. Nodes must be enabled to self-manage their configuration and they must be enabled to provide internetworking facilities between heterogeneously configured groups of nodes. Such self-organisation is premised on an ability to observe the pertinent state of the network, i.e. the networking conditions and the network composition, to analyse those observations and to make decisions, either on a unilateral or multilateral basis, to correct the network. Most importantly, any techniques used to effect these aims must ensure that the robust, distributed, infrastructure-independent and open essence of mobile wireless ad hoc networks is maintained.

To enable the desired flexibility a multi-protocol network-layer that facilitates node auto-configuration, node reconfiguration and ad hoc internetworking is designed. The auto-configuration protocol uses a combination of stateless auto-configuration techniques used in the area of address auto-configuration and low-cost localised self-stabilisation techniques to maintain a stable and legal global state with regard to the configuration of the nodes' network-layers. The node reconfiguration protocol incorporates a more flexible approach to distributed consensus with a modified version of an innovation-decision process to provide a solution for network wide conflict free node reconfiguration. The ad hoc internetworking protocol makes use of a Broker that sits in boundary nodes between different ad hoc network regions. The Broker is responsible for ensuring that the scalability of the internetwork is maintained by throttling the rate at which control packets pass from one network into another. The Broker component is also responsible for transforming packets as they pass from one region into another so that they may be processed in the receiving region.

1.5 The Thesis

That it is feasible to design a flexible system at the network-layer of an open, mobile, wireless, ad hoc network that enables autonomous nodes to configure themselves to use a routing protocol, chosen from a suite of available routing protocols, which is suited to the prevailing network conditions that they observe. It is feasible to design such a system in a manner that preserves the inherent optimised characteristics of the underlying ad hoc protocols by using signalling techniques which are compatible with ad hoc routing protocols in use. Furthermore, it is feasible to design a flexible system that allows a network to configure itself on a regional basis when networking conditions are not globally uniform and when one protocol is not optimal for the entire network such that regional variation in the configuration of the network does not prevent nodes situated in heterogeneously configured regions from communicating on an end-to-end basis in a multi-hop peer-to-peer manner.

1.6 Contributions

In addressing flexibility in ad hoc networks through focusing on the network–layer the following contributions are made:

(1) A comprehensive review and analysis of ad hoc networking, focussing extensively on the routing protocols used in such networks, is presented. The review shows that ad hoc routing protocols are strongly affected by their networking conditions and that a one-size-fits-all approach cannot be adopted. The need for a reconfigurable, heterogeneous network-layer is motivated.

(2) The concept of flexibility and adaptability in ad hoc networks is developed through the re-engineering of the concept of the network-layer for mobile ad hoc networks. The specific requirements of such a network-layer are enumerated.

(3) Novel techniques to enable the local, passive observation of networking conditions pertaining to the performance of ad hoc routing protocols are designed and developed.

(4) Novel techniques for enabling the auto-configuration of ad hoc nodes are developed. These techniques ensure the stable and consistent configuration of a distributed multi-state networking system by providing for the robust and lightweight just-in-time correction of distributed network variables..

(5) Novel techniques for enabling the reconfiguration of ad hoc nodes are developed. These techniques enable a mobile ad hoc network to reconfigure its variable parameters in response to the networking factors that affect its performance efficiency.

(6) Novel techniques for enabling the ad hoc internetworking of heterogeneously configured ad hoc domains are developed. These techniques facilitate end-to-end, multi-hop, peer-to-peer networking in an internetwork of dynamically reconfigurable heterogeneous routing domains.

1.7 Thesis Overview

The rest of the thesis is presented in eight chapters and two appendices that focus on the contributions listed in the previous section. The main thrust of each chapter is briefly described here.

Chapter 2 provides a review of ad hoc networks, looking at the challenges they pose and the variety of technologies that enable them. Particular emphasis is then given to the ad hoc routing protocols that are fundamental to the operation of multi-hop ad hoc networks. The body of work that has been carried out on the evaluation and comparison of ad hoc routing protocols is extensively reviewed. The implications for the future form of ad hoc networks that may be drawn from this work are then summarised. The influence of the needs of future networks are also reviewed and the relationship between emerging telecommunications' technologies and the urban space is discussed. This chapter provides a clear motivation for the design of an innovative flexible multi-protocol ad hoc networking system.

Chapter 3 presents the core idea of the thesis. The chapter firstly explores the the form that flexibility at the network-layer of an ad hoc network should take. This chapter introduces the concept of a network-layer built around a suite of ad hoc routing protocols that it may employ, as appropriate, in the course of its existence as the networking conditions that affect the operation of the different protocols undergo significant, long-term changes. This chapter describes the core of a multi-protocol ad hoc networking system and lays out what would be expected of the network-layer in such a system. This chapter provides an overview of the functionality of such a flexible network-layer, describing the step-change in design from a monolithic and preconfigured structure to a flexible multimodal structure which introduces a large number of demands on the network-layer. Significantly, this chapter sets forth the idea of a dynamically reconfigurable, multi-state ad hoc network. The chapter introduces a number of distinct components which should be incorporated into the multi-protocol network-layer in order to enable such flexibility.

Chapter 4 further develops the core idea presented in Chapter 3, exploring some of the basic design principles on which such a flexible network-layer should be developed. In doing so, it reviews the legacy networking systems in order to leverage the experience gained from their development. This chapter lays out clear design principles which are adhered to in the subsequent chapters, the aim of these design principles being to maintain the unique distinguishing traits of the open, autonomous, mobile, wireless ad hoc network.

A basic blueprint to guide the design of the flexibility-enabling components is presented; this blueprint emphasises the core elements of any such component as observation, analysis and decision-making. Finally, the structure of the basic network-layer, which includes the three routing protocols: AODV, DSR and OLSR, some attendant data structures and the experimental network-layer packet is also described in this chapter.

Chapter 5 explores a node's ability to observe the networking conditions that affect its performance; a flexible node should be capable of observing the dynamics of the network that affect the operation and efficiency of the network-layer so that the node can adapt or reconfigure itself in response to changed conditions. In particular, this chapter discusses a node's ability to passively observe the underlying network conditions in a manner that respects the system assumptions and propositions discussed in Chapter 4. Furthermore, this chapter goes on to describe a set of experiments that have been carried out to identify more precisely which protocols operate best in which conditions, i.e. to identify the performance profile crossover points between the three candidate routing protocols under a sample range of networking conditions.

Chapter 6 describes a flexibility-enabling component which provides for the auto-configuration of the network-layer's routing protocol. This chapter introduces and explores the issues involved in enabling flexible nodes to freely roam throughout a heterogeneously configured network. This chapter characterises the degree to which an ad hoc node can appreciate the configuration of its local environment using the signalling generated by the primary network-layer components. The chapter then presents a detailed review and analysis of a number distributed computing techniques that inform the design of the auto-configuration component. The review covers ad hoc management schemes, address auto-configuration and self-stabilisation techniques and it helps to highlight the limits of some approaches and the possible application of other approaches within the confines of this networking system. Finally, this chapter presents a protocol that addresses the problems that are associated with network-layer routing protocol auto-configuration. The auto-configuration protocol is composed of two sub-protocols; the Stateless Configuration Initialisation protocol and the Configuration Conflict Detection and Resolution protocol.

Chapter 7 describes a component which enables the reconfiguration of the network-layer's routing protocol. This chapter builds on the work that is presented in Chapters 5 and 6. As the co-ordination of change of a dependent variable across a group of distributed nodes in a mobile ad hoc network system is far from trivial, this chapter presents a review of the technique of distributed consensus. However, since the characteristics of mobile ad hoc networks do not conform to the expected system norms for which distributed consensus protocols may be designed, this chapter then reviews the social science concept of the diffusion of innovations in an effort to deduce a more flexible decision-making protocol for a flexible multi-protocol network-layer.

Chapter 8 describes a component which enables the ad hoc internetworking of heterogeneously configured ad hoc networks; specifically, the internetworking component enables the route discovery, route maintenance and, most importantly, packet-switching services to operate across a network composed of multiple heterogeneously configured regions of nodes. The chapter demonstrates that existing internetworking approaches, which are built on the notion of hierarchical networks and fixed gateways, are not suitable for the ad hoc networking domain. The chapter then goes on to develop a component that extends the philosophy

of the existing ad hoc networking techniques to an ad hoc internetworking environment. The chapter presents a comprehensive internetworking technique that enables connected regions of nodes to discover routes that extend from one region into another in a completely ad hoc, distributed manner that is as scalable as the underlying ad hoc routing protocols themselves

Chapter 9 concludes the thesis with a summary of the key contributions that are presented. This chapter also discusses some possible avenues of research that follow from this work.

Appendix A provides a detailed description of the three routing protocols that are used in the experimental system described in this thesis; AODV, DSR and OLSR. *Appendix B* provides a description of the concept of self-stabilisation by local checking and correction.

2 Ad Hoc Networks

2.1 Introduction

The primary purpose of this lengthy chapter is to introduce, review and analyse ad hoc networks and in doing so to give a comprehensive overview of the relevant research that has been carried out in the area. A particular emphasis is placed on demonstrating the extent of the heterogeneity that exists in the field of ad hoc networking, focusing on the various forms in which ad hoc networks manifest themselves and also the diverse functionality of the ad hoc networks. The early sections of this chapter look at the challenges faced by ad hoc network services and applications as well as looking at some of the technologies that will enable the deployment of ad hoc network systems. The later sections of the chapter focus on the network-layer as it exists in mobile wireless ad hoc networks. As multi-hop routing in an ad hoc network is of particular interest in this thesis, a detailed survey of the prior work on the development, evolution, evaluation and comparison of multi-hop routing protocols is presented. A discussion and summary of this survey is then presented as the survey literature continually informs the work presented in the following chapters. Finally, the last sections of this chapter feature a discussion of the future context in which ad hoc networks are likely to manifest themselves in open civilian environments.

2.2 An Overview of Mobile Wireless Ad Hoc Networks

Ad hoc networks are not a completely novel technology. Packet radio networks that do not rely on pre-existing infrastructure were developed for military purposes in the 1970s by the United States Department of Defence Applied Research Projects Agency (DARPA) in such projects as PRNET [Kah78], [Jub87], and SURAN [Bey90]. In recent years the advent of more powerful and more portable computing

devices and the development of a range of wireless transmission media have spurred a renewed interest in ad hoc network development within many non-military focused research communities.

An ad hoc network is an autonomous collection of routers that have the ability to dynamically and rapidly form networks without the use of any centralised network infrastructure. The routers of an ad hoc network also act as the hosts in the network and may commonly be referred to as nodes. The mobility of the nodes means that the network topology may change rapidly and unpredictably over time. Ad hoc networks employ multi-hop routing schemes that allow each network to grow, reduce in size, partition or merge in real-time without referencing any central authority in response to naturally occurring events. The need for multi-hop routes occurs because communicating nodes may not be within direct transmission range of each other, relying instead on intermediate nodes to relay data along the route.

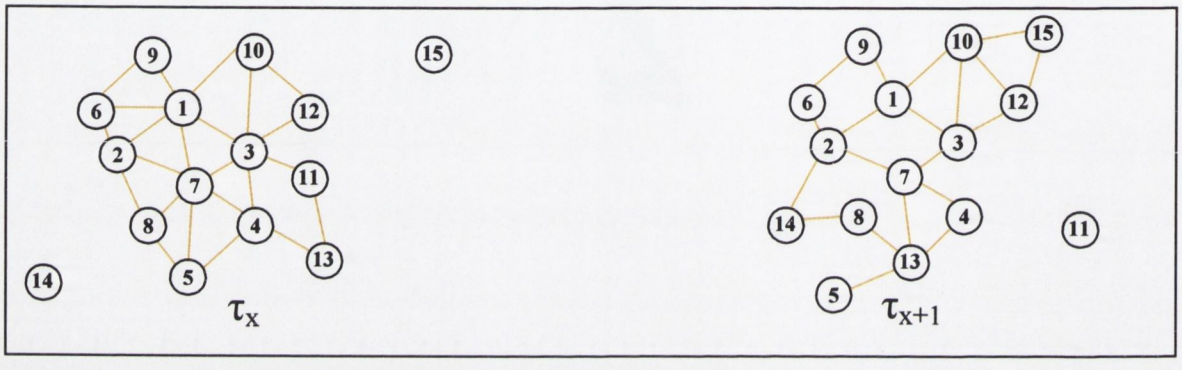


Figure 2-1 Nodes May Join or Leave an Ad Hoc Network at Any Time and at Will

Figure 2.1 illustrates a network of thirteen nodes with two unconnected nodes nearby at time τ_x. A short time later at time τ_{x+1} , nodes 14 and 15 have joined the network and node 11 has moved away. The topology of the network has substantially changed.

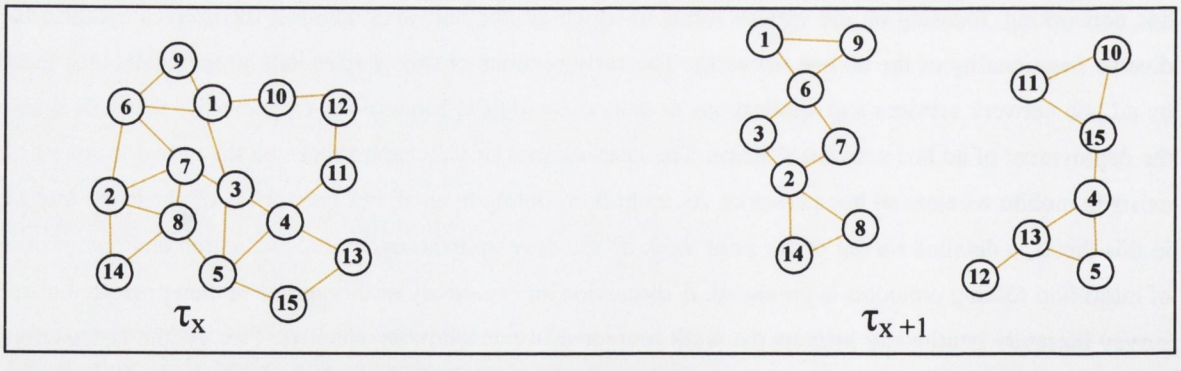


Figure 2-2 An Ad Hoc Network May Become Partitioned Due to the Mobility of its Nodes

Figure 2.2 illustrates a network initially at time τ_x and then a time later at τ_{x+1}. In the intervening time, seven nodes have moved away from the original network causing a partition to occur. Two separate networks now exist where formerly there was a single network.

2.2.1 Distinguishing Traits and Challenging Features of Ad Hoc Networks

Traditional infrastructure-based networks such as the Public Switched Telephone Network (PSTN) [Tan96] and the Global System for Mobile Communication (GSM) [Rah93] have centralised intelligence. The contemporary Internet, originally designed to combat the perceived frailties of centralised communications systems under the shadow of Cold-War era threats [Lyn93], promoted the use of packet-switched networks. Such systems are capable of dynamically and reactively re-routing packets around the network even when routers fail. A common trait of both the contemporary packet-switched and circuit-switched networks is the existence of, and reliance upon, powerful, static routers and switches. In such networks, the topography is planned in advance to cope with expected traffic patterns and to cover certain geographic locations. These types of networks take time to plan and deploy. They require a lot of manual configuration and intervention. In the case of GSM systems, it is necessary to plan the positioning of base station antennas to provide adequate geographical network coverage for end users. Ad hoc networks can be rapidly deployed as they do not rely on pre-existing architecture or any advance planning and human intervention.

An ad hoc network has a number of distinctive attributes: it operates in an autonomous fashion, it is a highly distributed entity, the nodes communicate via wireless links and tend to be energy constrained. These attributes bring both positives and negatives to the ad hoc network. A selection of attributes is described in brief:

Autonomous operation – The lack of restrictions imposed by centralised entities in an ad hoc network offers many advantages to the designer of an autonomous ad hoc network. For example, an autonomous ad hoc network may use its own mechanisms for accounting and authorisation. The network may also employ applications and services tailored to its environment, its users and its policies. On the negative side, the autonomous operation of nodes presents problems when designing protocols that must deal with unforeseen network changes and demands. Schemes may be required to enforce fair sharing of network resources through load-balancing, topology planning or payment schemes.

Distributed operation – The distributed co-ordination of tasks in any network of remote processes, or nodes, is a complex task. In an ad hoc network, the regular complexities of distributed algorithms are compounded by the dynamic network topology and the asynchronous nature of the nodes' communication and processing abilities. However, if these problems can be overcome, then the distributed nature of such a system makes it more robust and survivable in demanding scenarios as the inherent frailties of hierarchical systems are not present.

Wireless links - On the positive side, ad hoc networks may be placed in almost any situation as there is no need for wire or cabling. Wire has a significant cost factor for purchase and installation when compared to wireless solutions. The absence of wire means that ad hoc networks may be placed in sensitive environments without interfering with the physical fabric of buildings. However, as the transmission radius of each node is generally less than the span of the network, nodes must rely on the cooperation of other intermediate nodes to communicate with remote hosts and servers.

Energy constrained –Being mobile, nodes do not have continuous access to constant mains power supplies. As battery fuel cells have a finite supply of energy, nodes should act to conserve their energy reserves. To this end, ad hoc services must be designed to be judicious in their use of power draining activities such as extraneous transmissions and computations.

CPU and Memory constrained - Wireless ad hoc nodes will often be portable to enable node mobility. Portability is dependent on the weight and size of a node. Tanks may carry nodes weighing many kilograms whereas humans will not tolerate such weights for long periods. The miniaturisation of devices, to address weight and size concerns and to enable portability, generally results in the device having to compromise on the amount of space and weight it can give to the processor, memory and associated hardware. The computing resources of the average mobile wireless ad hoc node will fall far short of the resources of a typical 40Kg Cisco Internet router.

Low network capacity – Ad hoc networks rely on wireless links for communication. Wireless links are error prone and offer lower bandwidth than the wired links found in contemporary networks. The lack of bandwidth constrains the overall capacity of the network [Li01]. This is another factor motivating the judicious use of the wireless links.

The future form of ad hoc networks is unclear. As outlined, ad hoc networks have many distinguishing features that differentiate them from traditional networks. These characteristics of ad hoc networks have been found to have an impact on the many layers of the communications stack, not least the network-layer. It is informative to look at the ways in which these characteristics have impacted the layers above and below the network-layer, such as the medium access control (MAC) layer and the transport-layer. The direction in which these layers are being developed serves to illustrate the manner in which ad hoc networking technologies are diverging from the traditional Internet-type technologies and client/server network structures.

Typical MAC protocols that have been developed for wireless networks, such as the MACA [Kar96] and MACAW [Bha94] schemes are based on contention-free or contention-oriented mechanisms that enable the sharing of a single wireless channel and the avoidance of packet collisions. A comprehensive survey of many MAC protocols is presented in [Cha00]. In the present development of ad hoc networks, the IEEE 802.11 DCF MAC [Jor02] has been a popular choice. This scheme was developed to provide for distributed control of one-hop connectivity over the IEEE 802.11 radios. It was not originally envisaged for use in large-scale multi-hop wireless networks. The interaction between the MAC-layer and the ad hoc network-layer has been studied in [Roy00b], [Xu01], [Li01], primarily focusing on the effects of the IEEE 802.11 DCF MAC scheme on the performance of the ad hoc network-layer. [Li01] has found that, depending on the pattern of the routes used in the network, the capacity of the network can be significantly reduced by the interaction of adjacent nodes contending for the shared medium. However, the work presented in [Roy00b] has investigated the operation of a selection of routing protocols over different MAC schemes and found that the relative performance of the evaluated protocols did not show notable variation over the different MAC schemes.

[Xu01] contends that MAC schemes need to be specifically designed for the mobile ad hoc networking environment and that the existing batch of MAC schemes, specifically the IEEE 802.11 DCF MAC, were not designed for use in this scenario. The negative attributes noted in [Xu01] particularly affect the transport-layer. The interaction of the MAC protocols and the Transport Control Protocol (TCP) [Poe81] scheme in wireless ad hoc networks has been explored in [Ger99], [Tan99], and [Tan99b]. All of these studies have noted a strong negative interaction between the two layers in mobile ad hoc networks, especially when there are competing TCP streams. TCP has been studied as variants of the TCP family of protocols are used in over 95% of the fixed Internet.

As TCP has been designed for reliable fixed networks, it misinterprets packet loss as congestion and it invokes congestion control mechanisms that lead to retransmissions and loss of throughput [Cha01]. [Cha01] and [Yan03] have sought to modify existing TCP schemes for use in mobile ad hoc networks, while [Sun03] proposes a completely new reliable transport scheme for ad hoc networks which they have shown through simulation to outperform TCP.

The interaction of the MAC, network and transport layers of mobile wireless ad hoc networks demonstrates that the distinguishing traits of this type of network require developers to address anew the solutions that are used in the existing topologically stable and generally wired Internet. At the higher layers developers must redesign services such as name resolution, file system management and mail and web services. Many of these types of services are built on the client/server model which is unsuited to the mobile ad hoc networking environment. These services are generally configured for an organisation's environment by human administrators or automated tools such as the Dynamic Host Configuration Protocol (DHCP) [Dro97]. Developers must counter the challenges brought about by the network being unplanned in nature and having poorly defined boundaries, the lack of central servers, the absence of host configuration mechanisms and the fact that most users are not expert enough to manually intervene in a node's set-up [Fee01].

2.2.2 Applications of Ad Hoc Networks

Ad hoc networks may be used in a variety of settings for many different purposes, ranging from military tactical communications networks to free community networks in urban environments. An ad hoc network may be an isolated stand-alone network, having no connection to other networks such as the Internet or other ad hoc networks, or it may be used to extend the reach of existing networks [Tse03], [Per96]. The following list is just a small sample of suggested uses for ad hoc networks:

- Military/Government
  - Large scale military networks such as when forces are being rapidly deployed in unknown and hostile terrains.
  - Ad hoc networks may also be used by governments in domestic settings when the existing infrastructure is inadequate or compromised due to terrorist attacks or natural disasters.
  - Reconnaissance sensors may be deployed to gather data from remote terrains in preparation for action in those terrains [Est99], [Kah02].
- Education

- Students may use electronic resources in lecture theatres and cooperate on projects using ad hoc networking services.
- Campus networks may use ad hoc networks to automatically respond to changing needs as students congregate in different venues such as bars, cafés and lecture theatres.

- Community
  - Ad hoc networks may extend the HAM radio experience into the digital age.
  - Authority-free community networks may be built in urban areas where inadequate services are provided by the incumbent telecommunication companies.
- Disaster Relief
  - Ad hoc networks may be used for relief operations where existing infrastructure does not suffice – e.g. September 11, 2001, when communication systems failed, compromising rescuers abilities to complete tasks. Ad hoc networks may be used to enhance capacity and will adapt to changing and unknown terrains.
- Commercial
  - Exhibition areas may use ad hoc networks to enhance the exhibition experience by providing information services customised for exhibits, specific users and different locations.
  - Conferences and business meetings may use ad hoc networks to allow delegates to share personal and business information, to market and demonstrate products or to give impromptu presentations.
  - Office and desktop ad hoc personal area networks (PANs) may be set up using ad hoc networking technologies.
- Extending Infrastructure Networks
  - The range of the fixed network may be extended into areas that are not served by sufficient networking capacity, whether for commercial, regulatory or geographical reasons [Xu03].

While the military application of ad hoc networks has been the driving force for research since 1970's, the large scale use of ad hoc networking technology will be dependent on their applicability to non-military settings. While the range of possible applications for ad hoc networks is quite varied, no definitive, or so-called 'killer', application has been identified for the non-military sector. The list of scenarios that has been presented is by no means exhaustive or particularly imaginative. It is therefore a difficult feat to target the design of an ad hoc node to a service or task that is unknown.

2.2.3 Enabling Technologies for Ad Hoc Networks

As has been observed, ad hoc networks have both hardware and software components. In both cases a variety of options exist for the creation of functional ad hoc networks. In the military domain, the development of proprietary computing platforms and radio front-ends are within those organisation's budgets. Robust and portable computing platforms may be developed and flexible radio systems, that are adaptive to the needs of applications and the surrounding spectral activity, may be designed.

In the short term, ad hoc networks in the commercial arena are likely to be based on non-dedicated, or off-the-shelf, commercial devices. The proliferation of technologies that enable ad hoc networks in civilian environments increases the possibility that a market for ad hoc networking technology may exist outside of the military and research community domains. The growth of the personal computer market in the 1990s had an expansionary and commercialising effect on the development of the conventional Internet [Cof02]. It is therefore useful to evaluate how existing commercial devices can be used to create ad hoc networks. Initial attempts to push ad hoc networks beyond the laboratory door are very likely to utilise contemporary commercial wireless technology and computing platform options.

A sample selection of the current wireless technologies are reviewed to demonstrate the diversity of form and function that currently exists in this area: IEEE 802.11 [Jor02], Bluetooth [Jor02], ZigBee [Bah02] and UWB [Fre02]. Future ad hoc networking systems will either depend on these technologies for point-to-point connectivity or will have to exist alongside them. While none of these technologies are aimed specifically at the ad hoc networking domain, they may be adapted for use in this area.

I. *IEEE 802.11* is a family of standards that have been evolving over the last decade. IEEE 802.11 began as a wireless extension technology for conventional wired local area networks. It has since grown in both complexity and capability. The history and evolution of this family of standards may be found in [Jor02]. The various physical-layer standards that have emerged since have found varying levels of favour in different markets. Regulatory issues and political protectionism have inhibited the success of IEEE 802.11a in Europe, in a failed effort to allow the European HiperLan standard to flourish. HiperLan has not achieved the critical level of market saturation needed to make it accessible or desirable.

The most popular version of the standard so far has been IEEE 802.11b. This standard, commercially better known as WiFi, has taken off in the last couple of years [Bla02]. So-called WiFi 'hotspots' have also been growing in numbers in the last few years. The U.S. coffee house chain, Starbucks, now offers WiFi connectivity in many of its U.S. stores. In Dublin, the international airport, intercity train stations and a selection of public houses are now offering WiFi connectivity, albeit at inhibitive prices. Nokia, recognising the possible synergies between GSM and WiFi, has developed a WiFi/SIM card that uses the GSM network for authentication. In New York City, the provision of WiFi hotspots in public parks and spaces has been steadily growing in recent times.

IEEE 802.11b hotspots use the infrastructure-based IEEE 802.11 MAC mechanism which is common to all IEEE 802.11 variants. This mechanism, the Infrastructure Basic Service Set (BSS) uses access points to control communication between nodes in a contention-free manner. Mobile nodes associate with an access point and it periodically polls them to see if they want to communicate. The Infrastructure BSS is generally used for relatively long-lived communication. The second option is called the Independent BSS and allows mobile nodes to communicate with other nodes that are within direct communication range using a contention-based MAC mechanism. This mechanism supports short-lived communication sessions and is more suited to ad hoc networking scenarios as there is no hierarchy involved.

The next generation of the physical standard, IEEE 802.11g, offering much higher data rates than IEEE 802.11b, is being brought to market at the moment. However, the IEEE 802.11 MAC was not primarily

designed for use in multi-hop ad hoc networks that may span more than three or four nodes. Some of the limits that IEEE 802.11 may have on the performance of ad hoc networks have been noted in the last subsection.

II. *Bluetooth* is a short-range wireless PAN technology. It was initially developed as a cable replacement technology, but it has been developed into a standard for short-range voice and data networks. The Bluetooth standard specifies the technology from the physical-layer up to the applications and services. Different configurations of the Bluetooth stack are called profiles. Bluetooth profiles range in ability from the simple cable replacement mechanism to the ad hoc networking of devices within immediate range of each other to form *piconets*. Piconets are star-shaped networks composed of a master and up to seven slaves. It should be noted that the Bluetooth piconet is a star-shaped network that enables ad hoc connectivity; it is not an ad hoc network as the form of the network topology is fixed. Slaves may only communicate through the master and piconets may not exist in any other form. Up to ten piconets may occupy the same space, but they operate on different channels. The concept of '*scatterneting*' has been developed to allow networking between adjacent piconets. [Beu03] presents a study of Bluetooth in a mobile ad hoc networking environment while [Ger01] compares the use of 802.11 and Bluetooth when used for indoor ad hoc PANs supporting traffic such as voice and video. The [Ger01] study found that Bluetooth did not suffer from the TCP-associated problems that can occur over IEEE 802.11. Bluetooth was found to provide a more stable service which suits video streaming. The Routing Vector Method [Bha99] is a protocol that has been developed for ad hoc routing in Bluetooth scatternets.

One of the main questions of interest to developers, spectrum regulators and manufacturers has been that of the compatibility of Bluetooth and IEEE 802.11. [Joh02], [Lan01] and [Sil01] investigated the question of whether 802.11 and Bluetooth are complementary or competing technologies. [Lan01] looked at the ability of the two technologies to coexist at the physical and medium access levels. The study investigated the levels of interference and degradation caused by the simultaneous collocated operation of the two radio standards. The study noted performance degradation in both systems with a fall-off in throughout for both technologies. [Sil01] looked at the two protocols from a user scenario and application point-of-view, finding the two technologies to be complementary based on a comparison of price, range and data rates.

III. *ZigBee* is a technology that aims to offer reliable, low-power wireless networking for the purposes of controlling and monitoring devices. It aims to fill a perceived gap between the services offered by Bluetooth and IEEE 802.11. The IEEE 802.15.4 group have standardised ZigBee, their primary goals being simplicity, long battery life, reliability and cost. The target applications of ZigBee are in control and monitoring devices such as fitness monitors, TV remotes, lighting controls, keyboards and other home/office automation tools. These are devices where the battery is rarely replaced and ZigBee aims to extend its battery life beyond two years. ZigBee has a peak data rate of 128 Kbits/s and has a network join time of about 30ms, making it suitable for time critical applications. The network join time for Bluetooth is more than 3s. ZigBee supports up to 255 devices in one network.

ZigBee's main competitor would be seen as Bluetooth; IEEE 802.11 is too expensive and complex for the proposed tasks. ZigBee focuses on more static networks, carrying smaller data payloads between infrequently

used devices. Bluetooth, on the other hand, carries larger payloads in more frequently changing networks and facilitates services such as file transfers, streaming audio and video. Bluetooth devices are regularly recharged and are normally to be found in devices such as printers, mobile phones and audio devices that are either chargeable or connected to a mains power supply. The application of ad hoc networking technologies to ZigBee chipsets has not yet been explored. However ZigBee may be suitable for commercial low-power sensor systems that may be enhanced through the exploitation of ad hoc networking technology in many built environments such as the home, office or factory.

IV. *Ultra-wide bandwidth (UWB)* radios [Win98] have been receiving a lot of attention recently in their application to many new wireless communication systems, from home entertainment systems to factory automation services [Str03]. The U.S. Federal Communications Commission (FCC) defines UWB as any signal that occupies at least 500MHz of bandwidth between 3.1 GHz and 10.6 GHz, overlapping the 5GHz band utilised by IEEE802.11a. A standardised reference design, of the form offered by the IEEE 802.11/15 groups, is not yet available and specific application profiles have yet to be developed for UWB in a commercial, consumer-driven environment. It may be hoped that the eventual specification offered by the IEEE 802.15.3a group for UWB radios will have the foresight to anticipate, and flexibility to accommodate, the needs of future radio-based services, wireless ad hoc networking being just one such service. However, as of late 2003 the main proponents of the two competing standards for IEEE802.15.3a were in a standoff that centred on politics, royalty payments, and FCC compliance. Both Intel/Texas Instruments and Motorola are proposing quite different standards, the former based on multi-band OFDM and the latter on Direct Sequence CDMA. While standardisation is not now foreseen until 2005, proprietary devices may be developed in the interim.

Formerly, the techniques employed by UWB radios were widely used in radar applications. The high resolution of the UWB signal enables devices to recover location information with a high precision. It has also been suggested that UWB has the potential to allow simultaneous communication between a large number of user at high bit rates [Nar02]. The authors of that study have investigated the use of UWB radios in the creation of ad hoc networks and have shown it to be feasible. Regulatory authorities such as the FCC have imposed severe limitations on UWB power density to avoid interference with other technologies using its frequency range. Low power will limit the transmission range of this technology, possibly making it suitable for ad hoc networking on the PAN scale. As the initial UWB-based products are being released into the marketplace, the FCC is due to reconsider the limits imposed on UWB when it can assess the real interference effects of UWB-based consumer applications on other wireless standards [Fre02].

In the context of civilian ad hoc networking, UWB radios have properties that make them good candidates for short range communications in dense multi-path environments such as may exist in urban environments. According to [Fre02], UWB offers a number of advantages to wireless communications systems. Low-power UWB is spectrally efficient, coexisting with narrowband signals. It offers high data rates, of the order of 20 to 100 Mbits/s at ranges up to 10m. Rates of 500 Mbits/s and 1 Gbit/s are potentially achievable at lower ranges. UWB demonstrates a strong resistance to negative multi-path effects, making it quite suitable for indoor usage or in built environments. UWB operates at the microwatt power levels, the low duty-cycle of

the pulses making consumption very low. While UWB transmitters are simple, receivers are more complex but may be implemented in CMOS making the cost of the circuits a lot less than other wireless systems.

Table 2-1 Commercial Wireless Transmission Options

	IC Cost	Data Rate (Mbs)	Range (m)	Frequency Band	Output Power (mW)
IEEE 802.11 a	TBD	54	20	5GHz	40-800
IEEE 802.11 b	$15	11	100	2.4GHz	200
IEEE 802.11 g	$30	54	50	2.4GHz	65
Bluetooth	$10	1	10	2.4GHz	100
ZigBee	Target <$1 by 2005	0.128	10	915MHz, 868MHz & 2.4GHz	10
UWB	2005 at the earliest	100-500	10	3.1-10.6GHz	1

It should be observed that the technologies mentioned, with the exception of UWB-based technologies, are first-generation technologies for the wireless personal and local area networking domain. As applications, services, chips, markets and regulatory authorities mature, both the form and the function of these technologies are very likely to become more suited to the prevailing user demands and expectations. The very idea of using the cable-replacement technology Bluetooth as a PAN technology illustrates the primitive stage of development of this area of wireless communications. The adaptation of Bluetooth to the more complex tasks now expected of it has been anything but smooth or successful.

As summarised in Table 2.1, the technologies in question have varying characteristics that will allow them to fulfil different roles at the lower levels of any heterogeneous wireless communications system. The cited technologies have very different price tags, target markets and proposed uses. While they all provide some form of wireless communication capability, it is at very different levels of range, data rate and complexity. The technologies fill different niches and needs in the projected market and are all likely to find some level of acceptance in their individual niches. This will inevitably lead to a future consisting of coexisting heterogeneous wireless technologies.

2.3 Multi-Hop Ad Hoc Routing Protocols

Ad hoc routing protocols are at the core of ad hoc networks as they allow remote nodes to communicate without resorting to primitive flooding techniques. Unicast multi-hop routing enables communication between distant nodes in a directed manner; the existence and availability of such multi-hop routing functionality in an ad hoc network forms the basis for most higher-layer peer-to-peer services and applications. The design and development of efficient multi-hop routing protocols in a distributed and dynamic environment is a non-trivial problem. There are many approaches to the design of this fundamental ad hoc networking service, the motivations for which are described here.

The conditions faced by protocols for mobile ad hoc routing are many and varied; the prevailing network conditions pose different challenges to the protocols in use. Whatever the networking conditions, an ad hoc routing protocol has a number of distinct goals. The *primary* goal of the routing protocol is to gather routing information in an efficient and expedient manner, using as few limited resources as possible. This goal is broken down into the subtasks of *route discovery*, *route maintenance* and, in most cases, *data forwarding*. *Secondary* goals for routing protocols involve such issues as low-latency routing, power-aware routing, route stability, load balancing and other quality-of-service (QoS) demands. More pressing considerations in the design of a mobile ad hoc routing protocol are the avoidance of routing loops, fast convergence after link failures, unidirectional link support and network partition support.

The range of applications for mobile ad hoc networks is diverse, ranging from small static networks that are constrained by limited power sources to large scale, mobile and highly dynamic networks. A discussion of possible applications was presented in Section 2.2.2. Regardless of the applications and services that are to be employed, ad hoc networks must employ efficient distributed routing algorithms. There are many factors that influence the design of a solution such as variable link quality, signal fading, multi-user interference, power usage and the rate of expected topology changes. Additionally, in more demanding situations, such as military scenarios or certain sensitive commercial scenarios, the preservation of certain levels of security is desirable and the minimisation of the probability of communication detection and probability of communication interception is of paramount importance.

The legacy of routing technologies in conventional wired networks has influenced the design of many ad hoc routing protocols. Many of the protocols suggested for use in ad hoc networks have leveraged the experience gained from the design and use of routing protocols such as the Routing Information Protocol (RIP) [Hed98], Open Shortest Path First (OSPF) [Moy98] and Intermediate System-Intermediate System (IS-IS) [Ora90]. RIP is a distance-vector-based routing protocol, [For62], used within homogeneous autonomous systems in the Internet. A distance-vector indicates the estimated cost, in time, hops, dollars or some other metric, of communicating with a given destination and the preferred outgoing path from the router to reach the stated destination. Routers exchange distance-vector information regularly to build up their view of the network's topology. As the number of destinations grows, the numbers of distance-vectors exchanged also grows. This approach is relatively slow to converge and slow to react to topology change.

OSPF and IS-IS use the link-state flooding approach, initially designed to overcome the perceived inefficiencies of the distance-vector approach. Routers estimate the state of the links with their immediate neighbouring routers and this link-state information is then periodically flooded throughout the network. The size of a link-state update is proportional to the number of routers in the originating router's immediate neighbourhood. When routers receive these link-state updates they are able build a complete view, in the form of a directed graph, of the network's topology. Edsger W. Dijkstra's shortest-path algorithm [Dij59] of 1959 is commonly used to compute the shortest path from a given source to a destination based on information in the graph. While this approach uses less bandwidth than the distance-vector approach, each router must do its own complex computations to construct a graph and execute Dijkstra's algorithm for a given route query. Both distance-vector and link-state algorithms have been advanced over the years for use on the conventional wired Internet, but they still fall far short of the requirements for a mobile ad hoc

network where router mobility and variable link quality bring about a relatively dynamic topology. RIP, IS-IS and OSPF were not designed to be very judicious in their use of network resources as such constraints are not inherent to their operational environment – the conventional wired Internet.

The Mobile Ad Hoc Networks (MANET) working group, which operates under the umbrella of the Internet Engineering Task Force (IETF), has been co-ordinating the investigation, evaluation and improvement of a number of competing and complementary generic ad hoc routing protocols with a view to standardising routing protocols that are suitable for use in wireless ad hoc networks with both static and dynamic topologies [Mac99]. A large number of generic protocols have been proposed to this body over the last number of years. (It should be noted that the IETF's MANET working group is not the only forum in which ad hoc routing protocols are currently being proposed, developed and standardised.)

Current proposals for routing protocols in mobile ad hoc networks, many of which are based on the approaches outlined in the preceding paragraphs, tend to fall into one of four broad categories; namely proactive, reactive, hybrid and location-based protocols. These classifications indicate the type of mechanism used by each protocol to determine routes to remote destinations. The continuum of routing protocol mechanisms runs from purely reactive protocols such as the Dynamic Source Routing (DSR) [Joh96] protocol at one end to purely proactive protocols such as the Optimised Link-State Protocol OLSR [Jac01] at the other end. Table 2.2, at the end of this section, summarises the properties of some of the more common ad hoc routing protocols.

Many different design approaches are taken to achieve the same performance targets, whether they are low latency, low control overhead or high throughput. Many protocols exist at varying points along the routing continuum; such protocols tend to borrow aspects from both the purely proactive class of protocols and the purely reactive class of protocols. Comprehensive reviews of the range of ad hoc routing protocols may be found in [Roy99], [Lee03], [Mau01], [Tse01] and [Cor03]. However, as the provision of flexible routing services is one of the central themes of this thesis a short summary of the existing body of routing protocol designs is presented here in order to clarify the differences between the varying types of routing methodologies available. A more detailed description of the Ad Hoc On-Demand Distance Vector (AODV) protocol [Per99], DSR and OLSR may be found in Appendix A as these protocols are studied in more depth throughout this thesis.

2.3.1 Proactive Protocols

Proactive protocols borrow heavily from the work that has been carried out previously in wired networks. These table-driven protocols tend to follow variations of either the link-state or distance-vector approach. Using periodic beacons, nodes gather neighbourhood information and then disseminate topology information throughout the network either periodically or in response to observed local topology changes. The main improvements that distinguish them from RIP, OSPF and the like are that these new protocols optimise the mechanisms used to flood the topology information throughout the wireless network. Optimising mechanisms quench the repeat flooding of packets, prevent routing loops and otherwise minimise the excess use of network bandwidth that is inherent to the old protocols, while still aiming to have the network

topology information converge quickly. Table 2.2 summarises the distinctive operational characteristics of a number of popular ad hoc routing protocols. As will be shown in Section 2.4, these protocol characteristics, such as the packet forwarding mechanism or the update triggers, influence the performance of the protocols under a range of challenging conditions.

The Destination Sequenced Distance Vector (DSDV) protocol [Per94] is based on RIP, but it introduces sequence numbers to avoid routing loops. The Optimised Link-State Routing (OLSR) protocol uses a multipoint relaying system to minimise the number of repetitions in rebroadcasting topology update packets. Other proactive or table-driven protocols, such as the Adaptive Distance-Vector (ADV) protocol [Bop01], the Core-Extraction Distributed Ad hoc Routing (CEDAR) protocol [Siv99], the FishEye Routing (FSR) protocol [Pei00], the Global State Routing (GSR) protocol [Che98], the Neighbourhood Aware Source Routing (NSR) protocol [Spo01], and the System and Traffic Dependent Adaptive Routing (STARA) protocol [Gup97] are generally modified versions of distance-vector and link-state routing protocols.

2.3.2 Reactive Protocols

Reactive protocols generally differ quite significantly from proactive protocols in that they do absolutely nothing until there is a demand for a route from a node. These demand-driven protocols react to requests for a route by flooding the network with requests for routing information to the specified destination. Route requests are responded to with route replies. Replies may come from either the requested destination node or from an intermediate node that has routing information about the requested destination in its route cache. The means by which request and replies are generated and processed differs from protocol to protocol. Some protocols accumulate the routing information as the request disseminates through the network, creating routes on an end-to-end basis. An example of this type of protocol is DSR. Other protocols build up the routing information in a distributed manner, creating routes on a hop-by-hop basis, e.g. AODV. Reactive protocols are able to minimise their use of network resources by only acting when requested by a higher-level layer or another router. If the network is idle, then these protocols throttle back to a minimal state of activity.

Reactive protocols include the Dynamic Source Routing (DSR) protocol, [Joh96b], Dynamic Load-Aware Routing (DLAR) protocol [Lee01], the Associativity-Based Routing (ABR) protocol [Toh96], the Ad Hoc On-Demand Distance Vector (AODV), the Signal Stability based Adaptive Routing (SSA) protocol [Dub96], the Source-Tree On-Demand Routing (SOAR) protocol [Roy01], the Temporally-Ordered Routing Algorithm (TORA) protocol [Par97].

2.3.3 Alternative Hybrid and Location-based Protocols

Alternative approaches to ad hoc routing include both hybrid and location-based routing protocols. Hybrid protocols seek to leverage the optimal aspects of both reactive and proactive protocols. The Zone Routing Protocol (ZRP) [Haa97] and Sharp Hybrid Adaptive Protocol (SHARP) [Ram03] are two such protocols that use a proactive protocol for routing within a limited zone of routers and a reactive protocol for routing beyond that zone.

Location-based protocols such as the Landmark Ad Hoc Routing Protocol [Ger00], Geographical Routing Algorithm (GRA) [Jai01], Grid Location Service [Li00] and the Distance Routing Effect Algorithm for Mobility (DREAM) [Bas98] rely on position information such as that which is available from the Global Positioning System (GPS) [Kap96]. The reactive Dynamic Source Routing protocol has also been adapted to use position information [Bas99]. Location-based protocols seek to take advantage of position knowledge to focus route discovery techniques on the area in which the destination node is thought to be and to anticipate changes in existing routes to help in the route maintenance process.

Table 2-2 Selective Summary of Ad Hoc Routing Protocol Features

	Protocol Classification	Forwarding Mechanism	Route Selection Metric	Update Trigger	Periodic Messages	Unicast/ Multicast	External Requirement
ADMR	Reactive	Multicast Forwarding Tree	Shortest Delay	Time	Multicast KeepAlives	Multicast	None
AODV	Reactive	Distance-Vector	Shortest Path	Time/ Event	Hellos	Unicast	None
DREAM	Location (Proactive)	Hop-by-Hop Location-based	Shortest Path/ Location	Mobility Dependent Time	Location Information	Unicast	GPS
DSDV	Proactive	Distance-Vector	Shortest Path	Time	Hello, Routing Updates	Unicast	None
DSR	Reactive	Source Routing	Shortest Path	Event	None	Unicast	None
FSR	Proactive	Link-State	Shortest Path	Time	Hellos, Routing Updates	Unicast	None
MAODV	Reactive	Multicast Forwarding Tree	Shortest Path	Time/ Event	Hellos	Multicast	None
OLSR	Proactive	Link-state and Multipoint Relays	Shortest Path	Time	Hellos, Routing Updates	Unicast	None

The design of multicast routing protocols for ad hoc networks has also led to the development of protocols such as the Adaptive Demand-Driven Multicast Routing (ADMR) protocol [Jet01] and the Multicast Ad Hoc On-Demand Distance Vector (MAODV) protocol [Roy99b]. While such protocols are not investigated or pursued in this thesis, they generally utilise mechanisms that are found in the aforementioned unicast protocols in cooperation with algorithms that allow for multicast trees to be constructed and maintained under ad hoc networking conditions.

2.4 Evaluation of Multi-Hop Ad Hoc Network Routing Protocols

It was noted in the last section that the number of proposed multi-hop routing protocols has increased in the last number of years. Each of the ad hoc routing protocols mentioned in the last section has been designed to overcome the difficulties of route discovery, route maintenance and data forwarding in response to the perceived challenges brought about by the variable characteristics of the network. Among these

characteristics are node speed, node degree, network size, traffic types, physical-layer properties and the size and shape of physical environment. The performance of many of these routing protocols has been evaluated and compared by many researchers in that time in an effort to highlight their strengths and also to expose any algorithmic weaknesses for necessary modification and improvement. Much of our understanding of the performance of wireless networks has been based on the simulation of these technologies on many different platforms. While simulators have their limitations, they are sometimes a necessary counterweight to the limits of engineering intuition and reasoning. The creation and evolution of ad hoc routing protocols and other services in an ad hoc network is often achieved through a mix of engineering knowledge and intuition and evaluation feedback. The difficulties of designing solutions for ad hoc networks are mirrored in the limitations of the simulators and test beds used to evaluate them.

In order to understand and appreciate the published work on ad hoc routing protocols presented in the Section 2.4.4, the core elements of the evaluation process are presented here, namely the types of evaluation platforms used, the models used to emulate networking environments and the metrics used to determine the fitness of proposed solutions. This overview demonstrates the difficulties in network simulation that mirror the difficulties in designing ad hoc networking solutions that will be robust enough to cope with real-world deployment.

2.4.1 An Overview of Network Simulation and Evaluation Techniques

The use of simulators in the context of the development, debugging and testing of applications and services for ad hoc networks is understandable given the difficulties in doing such research work on real networks. Such work, if carried out on a real network, would generally inhibit speedy progress through the initial stages of protocol development. Moreover, in the case of the development of services and protocols for wireless networks, the ability to create a real large-scale wirelessly enabled test bed for all stages of development, from early debugging to final performance tuning, is generally not within the limits of most researchers' time or financial resources.

The published evaluations have taken place on a variety of network simulators and test beds: *ns* [Fal03], MaRS [Ala91], [Ala92], GloMoSim [Zen98], CMU Monarch test bed [Mal99], [Mal01], APE [Lun02] and JEmu [Fly02]. MaRS and *ns* (Network Simulator) are discrete event simulators. *ns* was developed by the University of California at Berkeley [Fal03]. The simulator has been a popular choice in recent years and extensions have been made to it enhancing the types of network scenarios and services that it can model [Bre00]. The simulator was extended to allow for the evaluation of ad hoc networks by the Monarch research team [Bro98]. Due to the sheer difficulty and expense, both in time and money, that comes with experimentation on real test beds, the level of research that has been carried out on platforms such as the Monarch test bed or the APE test bed is more limited. The published experiences of the researchers using the Monarch test bed [Mal01] demonstrate the use of investing in such platforms. The Monarch group sited their test bed among buildings in a 700m by 300m area, using rented vehicles, laptop computers, WaveLAN radios and GPS location services to both create real ad hoc networking scenarios and to collect data that could be analysed later. However, the limited size and scale of the network described, coupled with difficulty of constructing such a network illustrates the difficulties of protocol development on such platforms. The JEmu

test bed was developed by the Network and Telecommunications Research Group at Trinity College Dublin. This test bed allows for the evaluation of networking scenarios in both simulated scenarios and also on a real test bed. The JEmu test bed allows a network under evaluation to be simultaneously split between the real test bed and the simulator. The JEmu emulator aims to reduce the amount of time and effort in testing routing protocols and other services in comparison with either full field trials or full simulation.

The simulation and emulation of ad hoc networking protocols and other services is not without its own complications. While the simulation of ad hoc networks has mainly been carried out over the last few years, the methods and tools used in this work have been around for some time as they have been used extensively in the simulation of wired networks and internetworks such as the Internet. It is therefore worthwhile noting the difficulties and caveats that the experience of Internet simulation has yielded when advancing to the simulation of wireless networks. While wireless ad hoc networks are quite different to conventional wired ones, the principles underlying the modelling and development of both systems are largely the same.

Sally Floyd, one of the leading researchers in the field of network simulation, and a developer of the *ns* simulator, has raised questions about the limitations of simulation in the development of networking protocols and services [Flo01]. She has noted that one of the difficulties of measurement and experimentation by simulation is that present simulators can only be used to explore protocol development within the context of the existing Internet. While they can be used to explore new environments, such as that found in the mobile wireless ad hoc networks domain, they cannot be used to explore new architectures for the future Internet. Floyd raises the point that results yielded by simulators are subject to change as the detail of the simulator's software implementation of the underlying models is changed. This point is pursued further in [Cav02]. Another problem with simulation is the definition of underlying models based on assumptions about the future size, scale and nature of the network. The explosive growth of the current wired Internet, at a rate of 80% growth per year since 1984, has stretched some applications and services to breaking point. As Floyd describes it, the current Internet is 'an immense moving target', having some key properties that make it exceedingly hard to characterise, and thus simulate. Floyd cites three key properties. The first property is based on the Internet Protocol (IP) architecture that allows for the unification of diverse technologies and administrative domains. While IP unifies and masks these domains and their differences from a user's point-of-view, it does not impose a uniformity of behaviour in the various domains. Modelling this diversity in a simulator is difficult. The second property that Floyd identifies is that the growth of heterogeneity within the Internet has gone beyond a scale that may be neglected, making the simulation of new services much more difficult. Any such difficulties that are true for Internet simulation become more exacerbated for the simulation of the new wireless mobile ad hoc networking domain, where the real scale, scope and application of the technology is still unknown and the possibility of the realisation of a heterogeneous ad hoc networking environment is quite probable. Finally, Floyd notes that the Internet has changed in drastic ways at certain points in its history. As noted in [Cof02], the arrival of the next widely popular application or service, the so-called 'killer app', may be viewed as a form of disruptive innovation as it possibly upsets any plans and forecasting that has gone before based on prior assumptions about expected applications and services. Web browsers took the Internet by storm in 1993 with the release of the Mosaic [Vet94] browser. Within 18 months, Web traffic became dominant on Internet backbones. Web traffic doubled every 7-8 weeks and continued to do so for about two years. Predictions about the size and scale of future Internet traffic proved to

be very wide of the mark. [Cof02] goes on to discuss the impact of such peer-to-peer applications as Napster and Gnutella and the SETI@home project. The author concludes that the most disruptive factor in network design is rapid and unexpected growth, whatever the underlying cause of that growth is. The unpredictability of change in the Internet, and its effects on simulation, are also highlighted by the changes that have occurred in the reliable transport-layer protocol. In recent years the SACK TCP and New Reno TCP have become the more dominant variants of TCP in the Internet. However, researchers are still proposing mechanisms to compensate for poor performance under Reno TCP, which was the dominant mechanism in use in the late 1990s, but which may have virtually disappeared from the Internet by the time that their proposals are deployed [Flo01]. The danger is that by designing protocols that are heavily influenced by or interact with current technologies or protocols, they may become inoperable in a short time as changes occur in those parts of the network fabric around which they were designed.

While a lot of the evaluation of ad hoc networks is done on these simulators, some have questioned the accuracy of such simulators and the models that are used in different simulation platforms [Cav02]. The authors of this work have taken three simulators; *ns*, GloMoSim and OPNET Modeler, and have compared the results yielded by these simulators after a series of standard flooding experiments that are available in each of the simulators as standard. The authors attributed the differences in the results to the differences in the manner in which each of the simulators models certain aspects of the wireless networking environment. The precise implementation of models differs from platform to platform and is often dependent on the underlying structure of the simulators processors. As the entire communication stack, from application layer to the physical channel, is modelled in such simulators, the errors or discrepancies in modelling may occur at any layer of the communication stack. One of the authors' recommendations is to implement all but the lowest layers of the stack in real code on dedicated hosts, leaving the simulator to deal with only the lowest layers. The JEmu simulator takes an approach similar to that recommended in [Cav02].

While the preceding paragraphs have presented a number of difficulties associated with the whole field of network simulation, they also highlight a number of important points about the general development of networking technologies. The Internet is a large and relatively stable network, but one which has nonetheless still exhibited dramatic changes in its short history. Traffic forecasts have regularly been proven incorrect. The pace of change in the development and evolution of ad hoc networks may be more rapid as ad hoc networks will be smaller in scale and are more suited to development as standalone networks. While the Internet is seen as an 'immense moving target', ad hoc networks may be seen as an array of smaller but rapidly moving targets. The ease with which inherently smaller and more independent ad hoc networks may be upgraded to use the latest protocols, services and wireless technologies means that it may be quite challenging to keep pace with them in the context of development through simulation.

2.4.2 Variable Evaluation Parameters

The ad hoc network is an adaptive and self-organising system in that it is designed to independently respond to changes in its networking environment. The variable *network dynamics* that describe the characteristics of the environment challenge the route discovery, route maintenance and data forwarding mechanisms of the routing protocols. In particular, the scenarios that are used in the evaluations are characterised by a number of

specific tuneable parameters: physical simulation space, node movement models, and communication models. These tuneable parameters are normally varied from one extreme to another throughout a series of experiments in order to determine the performance profile of a routing protocol.

I. Physical simulation area, transmission coverage area and network size

The physical simulation area, the transmission coverage area and the network size are three parameters that allow an evaluator to define certain simulation properties. The *simulation area* used in evaluations defines an area within which the nodes may move subject to the mobility model employed. The majority of researchers use either rectangular or square open spaces to model the user space. The size of the simulation space has impacts on the network in a number of ways, depending on the number of nodes under review (the network size) and their *transmission coverage area.* The term *network size* describes the number of nodes in a given simulation field.

The *node density* of a network defines the average area occupied by a single node in a simulation space. It is given by dividing the simulation area by the network size. The average neighbourhood size, the *node degree*, is given by dividing the transmission coverage area by the node density number. The transmission coverage area indicates the area covered by a node when it is transmitting a signal. Typically, a node has a footprint of $\pi.(\text{TxRad})^2$, where TxRad is the range of the node's radio. For example, given a simulation field of 300m by 600m, containing 30 nodes with transmission radii of 100m the following statements may be made. The network size is 30, transmission coverage area is $\pi.10000\text{m}^2$, the node density is 1 node per 6000m^2 and the node degree is approximately 5.23.

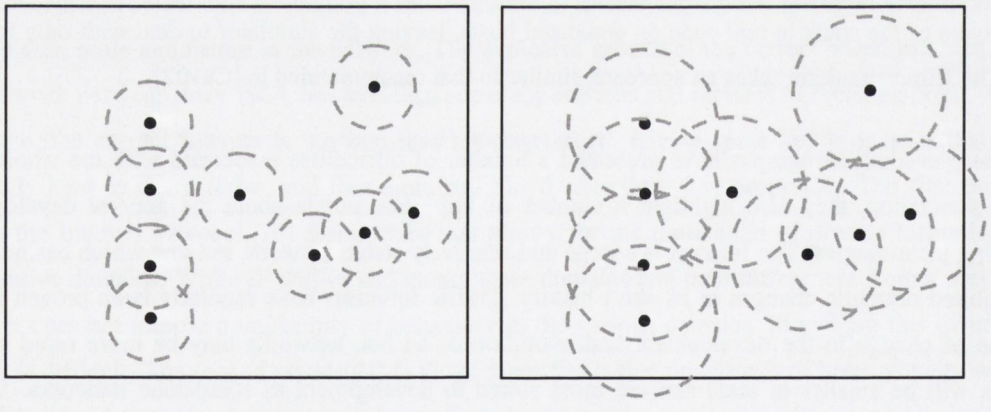


Figure 2-3 Two Networks of Nodes with Small Transmission Radii (left) and Larger Radii (right)

However, simply knowing the number of nodes in the network does not give a clear picture of the possible connectivity in that network. The network size information must be coupled with knowledge of the transmission radius of the nodes and the area of the simulation field. Figure 2.3 illustrates two networks occupying the same physical space and of the same network size. The transmission radius of the nodes in the first network is less than those in the second network. As may be seen, by comparing the two networks, there is more one-hop connectivity in the second network.

The maximum diameter of a network indicates the length of the longest route possible across the simulation area. It presupposes the uniform distribution of nodes over the network under a particular model of mobility. Longer paths are theoretically possible in simulation fields if nodes are strategically placed to maximise the path length. However, the maximum path length is only meant to roughly indicate a broad relationship between the simulation field and the transmission radius of the nodes under review. Figure 2.4 depicts a network with a theoretical maximum diameter of 8 hops, i.e. in such a simulation field, where the nodes are randomly distributed: the longest path should be about 8 hops. The maximum path length gives an indication of the limit that should be expected in the size of paths discovered by the routing protocols. Theoretically, a path should not exceed the maximum path length, thus allowing the researcher to estimate the maximum end-to-end delays that may be expected.

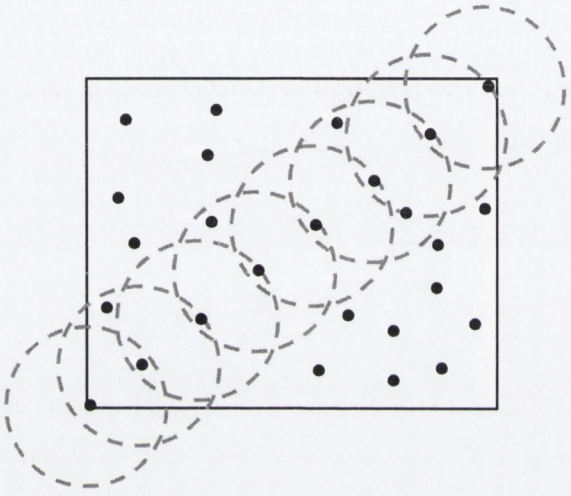


Figure 2-4 Determining the Maximum Diameter (in Hops) of a Network

In terms of the published literature that is reviewed in Subsection 2.4.4, evaluations tend to be typified by the use of low, moderate and high-density models. A low-density model is used in [Bro98] where the parameters have been set to yield average node degrees of 3.48 neighbours per node. Moderate density models are typified by node degree figures ranging from 7 neighbours per node to 12 neighbours per node, as are found in [Joh99] and [Das98]. The most dense, or well-connected, scenarios are typified by node degree figures of 15 neighbours per node [Per02] and 20 neighbours per node [Das01], and in one extreme case the average node degree is 50 neighbours per node [Joh99].

II. Movement models

The movement of nodes in simulated evaluations is governed by a mobility model, a mathematical model that creates an individual movement pattern for each node to follow for the duration of the simulation. Mobility models are used to control the movement of nodes, which may be emulating anything from pedestrians to fighter jets, by dictating the velocity of each node for the duration of the simulation. The mobility of the nodes impacts on the simulation in a number of ways.

The mobility of the ad hoc network nodes directly influences the rate of change of the network's topology. The rate of change of the ad hoc network's topology places a proportional stress on the ad hoc routing

protocol as the protocol must attempt to maintain routes and forward data. In an event-driven proactive protocol, the changes in the topology brought about by node mobility will cause it to transmit more control packets, while in a timer-driven proactive protocol, the rate of change of links in the topology will have less of an effect. Reactive protocols are affected by changes in the topology if they have active routes that need to be maintained. The operation of ad hoc routing protocols in static situations is not as complex as in mobile situations. In mobile scenarios, nodes move in and out of transmission range and node-to-node links are continually broken and created. Node failures in static networks will occur less often, mainly due to power failures or hardware/software failure. These types of infrequent node failure will have less impact on the network topology.

[Cam02] reviews many of the popular mobility models. Most of the mobility models that are cited in the evaluation literature are variations on a random mobility model. The most common model used is the random waypoint (RWP) mobility model, which was introduced by the Monarch team [Bro98]. The RWP model is akin to a drunken walk, as illustrated on the left of Figure 2.5. Initially, all nodes are placed randomly within the simulation area. The variable parameters of such a mobility model are the minimum and maximum permissible speeds, the direction of movement and the pause time. The minimum and maximum permissible speeds allow for nodes to be classified by their mobility model speed range. Nodes representing pedestrians may have a minimum speed of 1 m/s and a maximum speed of 5 m/s, whereas a car in an urban environment may have a maximum speed of 20 m/s, see Figure 2.6. The pause time may be varied from a minimum of 0s up to a maximum time equal to the simulation runtime. A pause time of 0s implies that the node never rests, continually moving about the simulation space. A pause time equal to the simulation runtime implies that the node stays in the initial position assigned to it. The RWP model gives the appearance of nodes wandering around the simulation space, their restlessness determined by the pause time. The strength of the RWP model is that it provides a uniform mobility benchmark against which many protocols and services can be evaluated and compared. The effect of changing the pause time from 0s to the maximum value varies the stress on the network by changing it from a network of continually mobile nodes to one of static nodes. Figure 2.5 (left) depicts one node in the upper left quadrant, for the sake of clarity, at a starting time, τ. At the end of the simulation, at time $\tau + \tau_{sim}$, the node has changed its velocity twenty-one times, after following a RWP model.

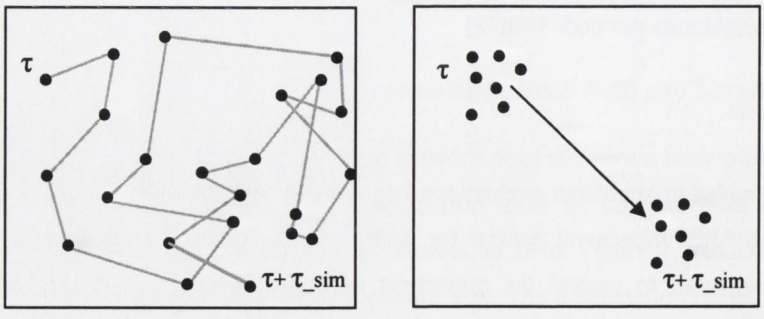


Figure 2-5 Sample Mobility Models: Random Waypoint (left), Nomadic Community (right)

Figure 2.5 (right) illustrates a Nomadic Community model. In this model, a network of nodes is depicted in the upper left quadrant of the simulation field at the starting time, τ. At the end of the simulation, at time $\tau +$

τ_{sim}, the entire network has moved to the bottom right quadrant of the field. This model may be used to simulate the movement of entire networks into different spaces; it may be used to simulate network mergers. However, it should be noted that the absolute mobility of the network does not affect the routing protocol, it is the underlying local relative mobility of the nodes that stresses the protocol routing mechanisms.

Some of the alternative mobility models that have been used include the Nomadic Community Model, the Pursue Mobility Model and the Random Direction Mobility Model. Models such as the Nomadic Community Model and the Pursue Mobility Model seek to mimic real life scenarios where groups of nodes move together to complete tasks, such as when a group of soldiers operate on a battlefield or a group of tourists roam through a city. Models may also be designed to specifically recreate naturally occurring events in ad hoc networks such as network joins, partitions and mergers. These types of models allow researchers to look at the effects of congregating and dispersing nodes, activities that may be more true to life than the 'drunken walk' of the RWP. However, in order to capture more realistic mobility scenarios, it is necessary to develop more complex models that tailor the mobility of the nodes to the proposed physical space and the task at hand. For example, the mobility of nodes in a train station cannot be modelled by any one of these protocols, but perhaps can be modelled by combining a number of the models to mimic the mobility of pedestrians on platforms, at ticket offices or in a spacious retail-oriented concourse.



0-1 m/s 1-5 m/s 6-20 m/s

Figure 2-6 Sample Node Speeds

In terms of the published literature that is reviewed in Subsection 2.4.4, evaluations tend to be typified by the use of low, moderate and high mobility models. Low mobility random waypoint models, which are typified by node velocity ranges of 0-1 m/s or 0.4-0.6 m/s and pause times of 0s to the maximum possible, i.e. the duration of the simulation, are found in [Bro98], [Joh99] and [Das98]. Moderate mobility models are typified by node velocity ranges of 1-5 m/s [Cla02b] and 3.5 – 4.5 m/s [Das98] with pause times varying from 0s up to the maximum possible. Finally, high mobility models are generally typified by maximum velocity ranges of 0-20 m/s [Bro98], [Joh99]. In an extreme case, [Per02], velocity ranges of 0-40 m/s have been used.

III. Traffic or communication patterns

The only basis for having an ad hoc network is to support the flow of data between clients and servers, sensors and data aggregators. The traffic that is loaded onto the ad hoc network both stimulates and stresses the route discovery, route maintenance and data forwarding mechanisms. Networks employing reactive protocols are particularly sensitive to the traffic that is loaded onto the ad hoc networks' nodes as the protocol adapts to the demands placed on it by traffic generating applications and services. The operation of proactive

protocols is not as sensitive to traffic loading as the protocols are generally active regardless of the applied traffic.

As with the other variable parameters of network simulators, the specific models used to generate traffic may be tuned to mimic certain types of real traffic events. The most common type of traffic used in evaluations is based on the use of constant bit rate (CBR) sources and sinks. A CBR source has three variable parameters; packet size, packet rate and packet destination. The CBR source emits packets at a constant rate. The rates used in simulations vary from 0.5 packet/s up to 20 packets/s. The size of the packets varies from a token 64 bytes to the maximum transferable unit (MTU) of the underlying link-layer protocol.

The number of sources, the number of streams, the source-sink pairings and the offered load stresses the system in different ways. The CBR sources and sinks may be set up in different ways to emulate different types of network traffic. The majority of published evaluations use CBR flows throughout the network to emulate client server interaction. It is generally a one-way flow. CBR sources and sinks may also be used to emulate sensor gathering systems or multicast-like information dissemination systems. CBR source-sink pairs may be spread about the network, the pairing may be static for the total simulation runtime or they may change at random intervals. Different scenarios may be modelled using varying configurations of CBR sources and sinks. One CBR sink and a lot of CBR sources may emulate a sensor aggregation application.

Some simulations use TCP to control data flows. However, TCP is not a popular choice as the protocol will throttle the offered load to the perceived throughput capabilities of the network. Essentially, TCP offers a conforming load to the network. It changes the times at which it sends packets based on its assessment of the network's capabilities. In order to compare the robustness and efficiency of different protocols, it is necessary to offer them the same traffic load consistently through the simulation regardless of their ability to deal with the traffic [Cla02c], [Bro98], and [Cla02b].

In terms of the published literature that is reviewed in Subsection 2.4.4, the traffic models used in the various papers defy strict categorization as they vary widely in terms of the number of sources used, the packet sizes and packet rates. Low CBR traffic loadings are typified by the use of 6 CBR sources in a population of 50 nodes where each CBR source sends packets at 4 packets/s [Bro98], whereas higher CBR traffic loadings are typified by the use of 25 CBR sources in a population of 50 nodes [Per02].

2.4.3 Performance Metrics

Having constructed environments to stress the route discovery, route maintenance and data forwarding mechanisms it is necessary to quantitatively evaluate the performance of the ad hoc routing protocol. In order to compare the relative performance of protocols under network conditions that are characterised by the simulation parameters outlined in the previous subsection, a number of standard metrics are defined.

The Advanced Wireless Technologies Division of the United States' National Institute of Standards and Technology (NIST) has suggested a selection of evaluation criteria to capture the performance of mechanisms employed in wireless ad hoc networks [Sub99]. It suggests that these criteria can be the determinants of the efficiency and effectiveness of an ad hoc network. The performance of ad hoc

networking protocols is generally described in terms of the following three metrics; average end-to-end delay, average throughput and average control overhead (also called the routing overhead or routing load). These metrics have been suggested by both NIST and the MANET working group as being suitable to assess the performance of any ad hoc routing protocol [Cor99].

- *End-to-end delay* measures the average amount of time elapsed between the time that the data arrives at the network-layer of the source node and the time that it subsequently arrives at the network-layer of the destination node.
- *Average throughput* or *packet delivery ratio* measures the effectiveness of the network in delivering data packets from the source to the intended destination. This is the ratio between the number of packets originated by the traffic-layer at the source and the number of packets received by the traffic-layer at the destination. This metric indicates the level of loss that would be seen by higher layers such as the transport-layer or application-layer. It is an indicator of the throughput that the network can support under the stated conditions.
- *Average control / routing overhead / routing load* indicates the average number of control packets produced per node. Control packets include route requests, replies, error messages and periodic beacons. Each transmission of a control packet, whether at the source node or an intermediate node, is counted only once. The routing overhead is an important metric as it is an indicator of the scalability of a protocol. The size of the traffic payloads that arrive at the network-layer from the higher traffic-layer may affect the overhead results for some protocols. The routing overhead may be measured both in bytes and in packets. Packet overhead may be significant where there is a per-packet cost to access to wireless medium.

Other performance metrics that are used in evaluations include *path optimality* and *power consumption*. As the first three metrics are common to almost all published evaluations, they are used for the purposes of comparison in the following discussion. It should be noted that the metrics used to analyse the performance of the protocols are dependent on the omniscient-observer, or god-like, abilities of the simulator. The simulator collates the data using its 'birds-eye' view of the events unfolding over the runtime of the simulation. However, in a completely distributed ad hoc network there is no such entity or agent at any node that would have access to such complete network performance data.

2.4.4 A Survey of Published Evaluations

A battery of evaluations has been undertaken since the resurgence of interest in mobile ad hoc networking technologies: [Bro98], [Cla02c], [Cla01], [Cla02b], [Das98], [Das01], [Joh99], [Lao02], [Lee99], [Per02], [Raj00], [Raj01], [Sug00], and [Vie03]. While the generic routing protocols mentioned in Section 2.3 were designed with the aim of gathering routing information in an efficient and expedient manner in an ad hoc network, they were often specifically tailored to certain types of networks. The authors of the OLSR algorithm, described in [Jac01], state that their proposed protocol is optimised for a specific set of network conditions: namely that it is "*well suited for networks, where the traffic is random and sporadic between a larger set of nodes rather than being almost exclusively between a small specific set of nodes*". The authors of TORA [Par97] designed their protocol "*for use in large, dense, mobile networks*" that operate in a "*highly*

dynamic network environment". The purpose of the evaluation studies, cited above, was to attempt to test the validity of the design claims through a series of varying and challenging experiments.

As discussed in the previous subsection, there are no standardised evaluation mechanisms and processes. The authors of many of the published evaluations have taken different evaluation approaches by varying the mobility and communication models or by changing the dimensions of the simulation space and varying the network size. The mix of approaches that have been taken to the process of ad hoc routing protocol evaluations makes the direct comparison of papers and results difficult. It should be noted that there are no unique standardised implementations of the routing protocols which are studied in the reviewed evaluations, a problem that has been noted by Floyd [Flo01]. These protocols are still in different stages of development and are therefore subject to change. Some researchers have chosen to implement their own version of particular protocols from the contemporary specification documents. Successive protocol specifications may include modifications to the core protocol mechanisms; modifications that have been brought about as a result of the evaluation process. Others have used protocol implementations that were already developed by other teams. In all cases the particular configuration of the protocols may differ from study to study as the protocols' fine tuning parameters are set to different values. Parameters that may be tuned in protocols vary from the interval for neighbour sensing beacons to the throttling mechanism used to stem the flood of route requests originated by reactive protocols. However, in the main, the overwhelming essence of each protocol is not compromised through their independent implementation. Indeed, a seminar has taken place to investigate the interoperability of independent AODV implementations [Roy02a]. The largely successful trial involved testing the interoperability of six implementations from institutions representing both industry and academia using a series of simple tests that culminated in multi-hop wireless routing. Minor bugs in some implementations were fixed on the day after some testing, illustrating that the initial discrepancies in implementations were quite minimal.

A sample selection of the published evaluation literature is reviewed here. For each of the eight examples of evaluation studies that are reviewed here, a short description of the simulation parameters is provided along with a discussion of the more pertinent findings of each study. Some of the main results, remarks and conclusions that may be drawn from the entire body of cited works are then summarised in the succeeding subsection. Exact details of the parameter settings and more precise descriptions of the simulation set-ups, along with graphical representation of the results, are to be found in each of the cited studies. For the sake of clarity, a very brief synopsis of the prominent protocols featured in the survey is presented.

AODV is a distributed on-demand hop-by-hop distance-vector based protocol. A node that requires a valid route to a destination floods a route request throughout the network, subject to a flood-quenching mechanism. Intermediate nodes that process the route requests add distance-vector based routing information to their route cache as the request disseminates through the network, creating a reverse-path for the reply. A route reply is unicast back to the source by a node that has valid routing information, either the destination node itself or an intermediate node, along the reverse-path. A forward-path is created at each node as the reply is processed and forwarded back to the source. Data is then forwarded to the destination on a hop-by-hop basis using the information about the recently discovered route that is distributed among the intermediate nodes.

Routes are locally maintained by means of local Hello beacons. If a route fails it may be repaired locally and route error messages are sent to nodes that were using the failed section of the route.

DSDV is a proactive distance-vector based protocol that is based on the Bellman-Ford routing mechanism [For62]. One of the main modifications that were made to adapt the Bellman-Ford algorithm for wireless networks was the introduction of sequence numbers to ensure freedom from routing loops. As a distance-vector protocol, every node in the network maintains a routing table containing an entry for every other possible destination node, along with the estimated number of hops to that node and the next-hop node. Each destination entry is marked by a destination-assigned sequence number that allows nodes to distinguish between fresh and stale routing information. The routing information is periodically updated by transmitting either full or incremental packets of routing information. Full dump packets contain all of the routing information stored in a node, while incremental packets only contain routing information that has recently changed. DSDV also floods updates when significant topology events occur, i.e. when links fail, meaning that DSDV is not a purely timer-driven proactive protocol.

DSR is a purely reactive protocol. A node that requires a valid route to a destination floods a route request throughout the network, subject to a flood-quenching mechanism. The route request packet accumulates route information as it disseminates through the network. A route reply, containing the entire source-route, is unicast back to the source by a node that has valid routing information, either the destination node itself or an intermediate node. The failure of a route is detected by the passive or active acknowledgement of packets being routed through the network. Routes may be locally repaired and route error messages are unicast to nodes that were using the failed section of the route. The protocol uses buffers to hold data pending the discovery or repair of routes.

OLSR is a proactive link-state protocol. Nodes may only route data for which they currently have a valid route. Data packets are not buffered while the routing information converges. The nodes determine link-state information by periodically beaconing their neighbours, allowing nodes to determine the quality of their neighbouring links. Each node then chooses an optimal subset of those neighbours to forward packets on behalf of the node. Routing information updates are then flooded through the network using the relay nodes to minimise the control overhead. Each node may then construct a complete graph of the network and may apply a route selection algorithm to that graph when a route to a destination is required.

TORA is a reactive protocol that uses a link-reversal algorithm to discover and maintain multiple routes to a destination. A TORA node that requires a valid route to a destination uses a height metric to establish a directed acyclic graph (DAG) which is rooted at the destination. A DAG will be created for each destination, if required. Intermediate links are assigned a direction based on the relative height of their neighbouring nodes; nodes may be either upstream or downstream. When a route breaks down a maintenance procedure re-establishes another DAG rooted at the destination. TORA assumes that all nodes have synchronised clocks, which is possible if the nodes have access to an external reliable resource, as the height metric is based on the actual time of the link-failures. TORA nodes engage a route erasure mechanism if they detect a network partition. The protocol may oscillate between DAG construction and route erasure if multiple sets of nodes are constructing DAGs and erasing DAGs based on information about each other. Unlike other protocols, it is

not guaranteed to converge. TORA uses the Internet Manet Encapsulation Protocol (IMEP) [Cor97] to provide reliable and ordered communication with its neighbours and to proactively sense the neighbourhood.

I. [Bro98] is an early paper that significantly contributed to the process of evaluating ad hoc routing protocols and introduced both the popular RWP model and the modified *ns-2* simulator. The modifications to the original *ns* simulator included the enabling of node mobility, the modelling of a physical-layer and the modelling of the IEEE 802.11 DCF MAC. This work compared the AODV, DSR, DSDV and TORA protocols using a range of standard tests. The authors of this paper have also designed and actively promoted DSR. The tests were carried out using 50 nodes in a simulation space of 1500m by 300m using the authors' RWP model with two speed ranges of 0-1m/s or 0-20m/s. CBR sources using rates of 1, 4 or 8 packets/s were used. The number of CBR sources varied between 10, 20 and 30. The node density, at 1 node per 9000m^2 combined with a coverage area of about 31400m^2, yielded an average node degree of about 3.48. These parameters were not varied for duration of each simulation exercise, which ran for 900s.

The authors focused on throughput, control overhead and path optimality as the key indicators of protocol performance. The last performance indicator, path optimality, is a seldom-used performance metric in this field of research. The path optimality metric indicates the difference in hops between the path that a packet took in the simulation, as discovered by the routing protocol, and the optimal path (or shortest path in this case) that was physically available at the time that the packet was originated. This metric gives an indication of the efficiency of a protocol in discovering and optimising paths under different networking dynamics.

In the first series of tests the maximum mobility of the nodes was held at 20m/s and the pause time was varied from 0s to a maximum equal to the simulation runtime. As may be expected, all protocols performed very well when there was no mobility, i.e. at a pause time equal to the simulation time.

The results for throughput indicate that this performance metric is only slightly dependent on mobility and almost totally independent of traffic load for both AODV and DSR. DSDV performs poorly at pause times lower than 300s, progressively getting worse as it fails to converge. DSDV does not maintain alternate routes in its route cache and a link break will result in all of the outgoing packets being dropped. TORA performs quite well with 10 or 20 sources but its performance really suffers when the number of sources is increased to 30. The failure of TORA to converge was found to be related to the congestion caused by its high control overhead.

As expected, the control overhead, measured in packets, for DSDV is shown to be independent of changing mobility and offered traffic load. DSDV is a proactive protocol that periodically broadcasts updates. The control overhead of AODV, DSR and TORA is, as may be expected of reactive protocols, found to be dependent on the mobility and offered load. Increased mobility and offered load increases the control overhead for both protocols. The rate at which the control overhead grows drops off for AODV and DSR as more sources are added since both protocols may use information learned during other route discovery processes. Nonetheless, DSR is found to outperform AODV by a factor of 4 due to its use of an aggressive route caching policy. The AODV route request propagation technique used in this study affected its performance negatively. DSR outperforms TORA by almost a factor of 10. TORA collapses under the

congestion caused by its proactive neighbour sensing mechanism coupled with its reactive route discovery mechanisms which use the reliable IMEP protocol [Cor97].

The simulations indicated that in terms of path optimality, DSDV and DSR outperform both AODV and TORA. The paths used by both DSR and DSDV are close to optimal. Both AODV and TORA are negatively affected by node mobility, using more optimal routes at high pause times. It should be noted that TORA was not designed to find the shortest path.

In the second series of tests that were carried out at a node mobility rate of 1m/s, all of the protocols performed well with regard to packet delivery. However DSR is shown to outperform AODV by a factor of ten, again due to its aggressive route caching policy. DSDV managed to converge in this scenario. Differences still emerge between the protocols when the control overhead, measured in packets, is examined. DSR now outperforms AODV by a factor of 10, twice that of the earlier scenario. This may again be explained by DSR's caching strategy. It is noted that if the control overhead was measured in bytes, then AODV would outperform DSR except at the highest node mobility rates. This is due to the source route mechanism used by DSR. DSR still performs better than both DSDV and TORA using this approach.

To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV**: More expensive than DSR at high mobility rates. Performs as well as DSR in all other scenarios and eliminates the high byte overhead associated with DSR. The control overhead of AODV is sensitive to increased traffic loading, exacerbating any negative effects brought about by mobility.
- **DSR**: Performs well at all node speeds. In byte terms, it performs poorly with regard to control overhead. Traffic loading affects DSR in a similar manner to AODV.
- **DSDV**: Performs well at low mobility rates when it can converge. DSDV performances are independent of both traffic and mobility. DSDV's control overhead remains high regardless of low traffic loading or node immobility. Overall, DSDV was found to perform poorly in moderately mobile situations, taking an excessive amount of time to converge. Specifically, DSDV failed to converge for pause times of less than 300s.
- **TORA**: It is the worst performer in all scenarios with regard to control overhead. High traffic loads have a negative impact on TORA's packet delivery performance.

II. The evaluation detailed in [Joh99] examined the AODV, DSDV and DSR protocols in a variety of standard tests and scenario-based tests on the *ns-2* platform using a 50 node population. The authors were among the first, of a small number of researchers, to experiment with simulation parameters that attempted to model so-called real-life scenarios. The standard tests were characterised by the use of 15 CBR sources operating at rates varying from 5-20 packets/s and a 0-20m/s RWP mobility model in a 1000m by 1000m simulation area. The node density for the standard tests was constant: 1 node per 20,000m^2 with a transmission coverage area of 196350m^2, yielding an average node degree of 9.8. The authors looked at end-to-end delay, throughput and control overhead as the performance metrics. Control overhead was measured in both packets and bytes.

The standard tests revealed that the end-to-end delay experienced by AODV and DSR increases with increased mobility, while AODV suffers more than DSR does. DSR has a lower delay than AODV due to its route caching policy. The link breakages associated with increased mobility will cause more delays in the route discovery and maintenance processes. This in turn leads to a delay in the forwarding of data packets from source to destination. This study notes a correlation between delay and node mobility for both the AODV and DSR protocols. DSDV exhibits low delay as only packets for which valid routes exist are sent. Packets, for which no route exists, are dropped. In DSR or AODV the packets would be buffered pending the discovery of a route, causing increased average delay.

As the traffic offered to the network increases, all protocols suffer with regard to their delay performance. This is due to the send buffers being filled faster than routes are discovered. AODV is the poorest performer at the highest traffic loads, but delivers more packets than its competitor protocols. The inability of DSDV to cope with higher mobility is exposed at higher loads as more packets are dropped. DSDV does not adapt the rate of its periodic updates in response to changing mobility.

The throughput of DSR and AODV was not found to vary significantly with respect to mobility, while it decreased for DSDV with increasing mobility. The throughput only decreases slightly for AODV and DSR with higher mobility. The low throughput for DSDV may be explained by its slow convergence. Higher mobility prevents the DSDV route discovery mechanisms from discovering topology changes in a timely manner, again due to its inability to react to such changes.

The throughput performance of DSR is not as good as AODV at higher loads as the DSR source routes tend to cause congestion at the nodes. The link-state routing approach used by AODV seems to make it more robust to higher loads and high mobility. DSDV suffers the highest drop-off in throughput at higher traffic loads as more packets are dropped on stale routes.

The control overhead of AODV and DSR was again found to be proportional to mobility. Increased mobility causes link breakages, engaging the route maintenance mechanisms in both protocols. This increases the control overhead. In byte terms, DSR has a much higher overhead than AODV. The byte overhead for DSR is more closely correlated to traffic load. Source routes also penalise DSR if the payload is small with respect to the source route size. AODV is more independent of traffic load as its control overhead is only caused by route discovery and maintenance packets. However, compared to DSDV, both reactive protocols incur far less overhead. DSDV does not adapt to changing mobility or traffic loads in any way.

A novel aspect of this evaluation paper was its secondary focus on scenario–based experiments. The authors looked at three different scenarios: an *event*, a *conference* and a *disaster*. Each scenario had distinct networking characteristics. The *event* took place in an area of 1500m by 900m and was populated by 50 nodes. The node speed ranged from 0-1 m/s, but with 50% of the nodes moving at any one time, this creates a very mobile environment. The nodes formed clusters of about 10 nodes from time to time, making short-lived short routes. The node density for the event scenario was 1 node per 27,000m^2, yielding an average node degree of 7.2. There were 9 CBR sources and 45 receivers, creating 45 streams of CBR data across the network. The scenario parameters lead to fluctuating traffic levels and fast changing topologies.

All three protocols managed to achieve high throughput performance in this scenario, however AODV incurs one tenth of the delay incurred by DSR. As the nodes were generally arranged in clusters this leads to the creation of short routes, often between neighbours. AODV's use of proactive HELLO updates on active routes helped to bring down the delay in cases where routes are often between neighbours. The byte overhead incurred by DSDV was ten times that of AODV and sixty times more than DSR. In packet overhead terms, both AODV and DSDV used between thirty and forty times more packets than DSR in this scenario.

The *conference* scenario exhibited low mobility and there were 6 CBR flows between 2 sources and 6 receivers in a population of 50 nodes. The conference area was 90m by 150m. The node speed ranged from 0-1 m/s, but only 10% of the nodes were moving at any one time, creating a low mobility scenario. The node density for the conference scenario was 1 node per $270m^2$ meaning that all nodes were theoretically within range of each other. It was a high density and low mobility scenario in which radio interference and congestion were to be expected. The stated purpose of this scenario was to test the ability of protocols to adapt to quickly to changes in long-lived routes as well as dealing with the effects of congestion. Both AODV and DSR outperform DSDV in this scenario with regard to throughput. While AODV and DSR have comparable levels of performance, AODV uses more than ten times as many control packets in this scenario. However, in byte terms, DSR is twice as costly as AODV in this regard.

The final mixed mobility model used for the *disaster* scenario was composed of a mix of human and vehicular nodes. The physical simulation area was the same as that for the event. The speeds ranged from 1 m/s to 20 m/s and there were 87 CBR flows. Three groups of low mobility humans were randomly connected via two high mobility vehicles moving between them. This scenario had the effect of creating temporary partitions in the network and then allowing for intermittent longer routes to be formed.

In this scenario, DSDV suffered under the effects of network partitioning, taking too long to converge when network connectivity was temporarily restored. However, DSDV had the best end-to-end delay profile for packets that it managed to send. This observation is typical of many scenarios. AODV outperformed DSR due to its proactive HELLO mechanism providing immediate routes to neighbours. In terms of control overhead, DSR suffered compared to AODV due to the existence of longer routes.

The scenarios used in this paper demonstrated the value of experimentation using models that somewhat mirror real situations. The scenario experiments enabled more distinctions to be made between the suitability of the candidate protocols.

The authors also defined a mobility metric that attempted to model and measure the mobility of the nodes based on the average of the individual mobility measures over time. The individual mobility measure between any two nodes is defined as their absolute relative speed taken as an average over time. This metric is made possible as the node velocity information for each node is available to the simulator engine. In reality, the collation of such information would be dependent on the existence of data aggregators and external systems such as GPS. Nonetheless, the authors' results indicate that there is an approximately proportional relationship between the rate of link formations and breakages and the relative mobility of the nodes.

To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV**: Generally, AODV performs well in terms of throughput, mobility having little impact on the results. As mobility increases however, so too does the end-to-end delay. The throughput of AODV is better than DSR at higher traffic loads as DSR source routes lead to more congestion, but the delay performance is not as good.
- **DSR**: DSR has a similar performance characteristic to AODV. They differentiate themselves in terms of the overhead that they incur. DSR generally uses fewer packets than AODV while the size of those packets is generally larger than those used in AODV. This is characteristic of the inherent differences between the distributed distance-vector routing and source-routing approaches.
- **DSDV**: The scenarios and standard tests used in this study illustrate the difficulties that a proactive routing protocol such as DSDV has in trying to cope with changes in the topology due to high mobility and moderate pause times. As DSDV updates are triggered by topology changes, increasing node mobility leads to increased control overhead. The scenario-based evaluations, using relatively low mobility models compared to the standard test, reinforced the poor perception of DSDV's ability to cope well relative to the reactive protocols.

III. The authors of [Per02] have taken a novel approach to the evaluation of routing protocols, choosing to investigate the network factors that affect the performance of the protocols. They pay particular attention to the interaction of the many variables in the networking environment. The experiments involved varying five network factors between two extremes. The factors were network size (which has the effect of changing the node connectivity or node degree), node speed, the number of traffic sources, node pause time and the ad hoc routing protocol itself. The RWP model was used to control the mobility of the nodes using the specified node speed and pause time parameters. The network size was either 50 or 80 nodes in an area of 1600m by 400m. The 50 node scenario yielded a node density of 1 node per 12800m^2 with a transmission coverage area of 196350m^2, giving an average node degree of 15.33, while the of 80 node scenario yielded a node density of 1 node per 8000m^2 with the same transmission coverage area giving an average node degree of 24.5. The maximum node speed was set at 5 m/s or 40 m/s while the pause time was set at either 3s or 30s. The number of CBR traffic sources was set at either 10 or 25, each source emitting packets at a rate of 4 packets/s. The two routing protocols used were AODV and DSR, AODV being a distributed hop-by-hop reactive protocol and DSR being a source routed reactive protocol. The authors primarily looked at throughput and control overhead as the indicators of protocol performance in this study. They also investigated the interaction of the variable scenario parameters in their affects on the performance indicators. It was generally observed that the pause time did not have a significant influence on any of the measured performances.

For control overhead, the authors observed that the main influencing factor was the speed of the nodes. The experiments also determined that AODV gave rise to more overhead than DSR when challenged by an increased node speed and number of traffic sources. An increase in the network size, i.e. the existence of more nodes in the same simulation area, was found to increase the throughput performance of both protocols. It was observed that for throughput, the routing protocols had almost identical performance profiles.

However, it was noted that high node speed and an increased number of traffic sources negatively impacted on both protocols' performances.

The results presented in this paper suggested that the dominant factors affecting the performance of ad hoc routing protocols are, in order of importance, the number of traffic sources, the speed of the nodes and the average node degree. The observation regarding traffic sources is not one that can be generalised for all protocols as traffic loading is a natural trigger for reactive protocols anyway, whereas proactive protocols are more insensitive to it.

To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV, DSR**: While the AODV protocol had the same delivery performance as DSR, it was found to incur more overhead, as measured in packets. This study did not identify any other striking differences between the two protocols, noting instead that other network properties had more influence on the performance results obtained.

IV. [Das01] investigated the performance of AODV and DSR, two reactive protocols. The authors of this study originally designed AODV and are active proponents of the protocol. However, this study provides one of the more in-depth comparative evaluations of these two prominent protocols. The networking environment was characterised by the use of a RWP mobility model having a maximum speed of 20m/s and a range of pause times. The communication model was based on 10, 20 or 30 CBR traffic sources sending packets at 4 packets/s. A scenario involving 40 CBR traffic sources sending packets at a rate of 3 packets/s was also used. Two simulation spaces were used. The first consisted of 50 nodes in an area of 1500m by 300m; the second consisted of 100 nodes in an area of 2200m by 600m. Given the communication ranges attributed to the nodes, the two simulation space configurations yielded average node densities of 1 node per $9000m^2$ and 1 node per $13200m^2$ respectively. The transmission coverage area was $196350m^2$, giving an average node degree of 21.8 for the first space and an average node degree of 14.86 for the second one. This setup allowed the authors to examine the effects of node mobility, node density and offered traffic load on the performance of the two protocols. The authors used the three most common performance metrics; end-to-end delay, control overhead and packet delivery.

In the first higher density scenario, the protocols were tested using different numbers of traffic sources and pause times. For the lower traffic loads, i.e. for 10 or 20 sources, both protocols show a comparable packet delivery performance. As the traffic load increases, under 30 and 40 sources, AODV outperforms DSR except at very low mobility. DSR loses up to 50% more packets than AODV at higher mobility rates.

The delay performance for DSR is better than AODV at low traffic loads and under high mobility conditions. As the mobility decreases the difference in delay is reduced. However, at higher traffic loads, AODV outperforms DSR for all mobility rates. The delay for both protocols increased with the heaviest traffic loading under very low mobility conditions. This is most likely due to congestion in the network. Traffic tends to follow the same paths in low mobility scenarios as there is no load balancing mechanism in either

protocol. This feature is not as pronounced as the mobility rates increase as the traffic is automatically distributed over new and changing routes.

In terms of control overhead, DSR consistently outperforms AODV. The difference in their relative performances increases with increased traffic loading. DSR's normalised control overhead is relatively stable with respect to the number of sources, increasing linearly as the number of traffic sources increases. This is a desirable quality as it indicates that DSR is scalable protocol with respect to offered traffic loads.

In the second lower density scenario, the protocols were again tested using different numbers of traffic sources and pause times. The delivery rates for both protocols are very similar under the stress of 10 traffic sources. However, as the traffic load increases AODV delivers twice as many packets for the heavier traffic scenarios regardless of the node mobility.

The performance of the protocols in terms of delay mirrors the delivery rate performances of both protocols. At traffic levels above that offered by 10 sources, the delay given by DSR is about twice that of AODV. The difference in the control overhead incurred by both protocols is not as pronounced in this setting. DSR consistently performs better by a factor of 1-3. In general though, AODV outperforms DSR, except under low traffic loads, with regard to the application oriented metrics; packet delivery and end-to-end delay.

In a final test, the effect of varying traffic loads was assessed using the highest mobility setting in the 100 node scenario. AODV has a consistently higher throughput and lower delay than DSR. The performance of DSR saturates earlier, under a lower traffic loading, than it does for AODV. To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV**: The author's protocol outperformed DSR in terms of delay and throughput except under the lowest traffic loading. AODV significantly outperforms DSR as the node mobility and traffic loads increase.
- **DSR**: DSR is the best performer in the least stressful scenarios, where the network size is smaller and lower traffic loads and node mobility rates are applied. DSR generates less overhead than AODV regardless of the scenario characteristics. The poor delay performance of DSR is attributed to its use of cached routes that may be stale, whereas the AODV expires links more often which allows it to freshen the route information sooner.

V. [Raj00] presented a comparison of DSR and the proactive WRP-Lite protocol using Nokia's C++ Protocol Toolkit platform. The authors are the designers of the WRP-Lite protocol. The networking environment for this study was characterised by a simulation field 6600 by 4800 containing 20 nodes. The traffic model was characterised by either 8 or 16 CBR sources emitting packets at a rate of 4 packets/s. The RWP mobility model was used with a speed range of 0-20m/s. The authors used delivery ratio, control overhead, path-length and end-to-end delay metrics to measure the performance of the protocols.

For the scenario involving the lighter traffic load of 8 CBR sources WRP-Lite and DSR displayed similar control overhead performances, with WRP-Lite performing slightly better at higher mobility rates. DSR

outperforms WRP-Lite when the nodes are static. DSR slightly outperforms WRP-Lite on packet delivery terms for all mobility rates. However, WRP-Lite is shown to use optimal paths at higher mobility rates, responding to link changes, and the availability of shorter paths, faster than DSR does. WRP-Lite has a much better delay performance than DSR. This may be explained, as for most proactive protocols, by the fact that WRP-Lite drops packets for which it does not have a valid path whereas DSR buffers them, causing delay, while discovering new paths. The delay performance figures do not compare like with like, which favours WRP-Lite.

In the scenarios involving the heavier traffic load caused by 16 CBR traffic sources, the control overhead for DSR is shown to be an order of magnitude worse than for WRP-Lite. The WRP-Lite control overhead performance does not change substantially from the earlier scenarios involving a lighter traffic load. The delivery performance of the two protocols is very similar with DSR only suffering at the highest mobility rate, a pause time of 0s. Again, the delay performance profile of DSR is not as good as WRP-Lite. The aforementioned reasons for this performance still hold.

To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **DSR**: The delivery performance of DSR is very similar to that of WRP-Lite. While DSR suffers at the highest mobility rates, it is able to throttle back its control overhead in more static situations, unlike WRP-Lite.
- **WRP-Lite**: Performs well with regard to delay and hop distribution irrespective of traffic demands. The comparisons for delay are misleading as they represent two different types of service which may be suited to different applications. WRP-Lite generally has a lower control overhead profile.

VI. The authors of [Das98] compared the performance of five protocols: AODV, DSDV, DSR, the Extended Bellman Ford (EXBF) protocol, the Shortest Path First (SPF) protocol and TORA. The implementations of both AODV and DSR in this study were not optimal; neither protocol had recourse to send buffers and neither protocol implemented route request quenching techniques. The absence of such features limits the efficient operation of both protocols in accordance with their stated design. Using the MaRS platform, the networking environment for this study was characterised by 30 nodes in a 1000m by 1000m simulation field. The node density is 1 node per 33,333m2 with a transmission coverage area of 384650m^2, giving an average node degree of 11.5. The authors used a form of the RWP model similar to, but not the same as, the RWP model defined in [Bro98]. The minimum and maximum node speeds were set at 0.4 m/s – 0.6 m/s and 3.5 m/s – 4.5 m/s respectively. The authors also used a form of CBR traffic with packets sent at intervals having an exponentially distributed mean of 300 ms. The performance of the protocols was measures using the control overhead, delivery rate and end-to-end delay metrics.

The control overhead characteristics determined by this study confirm some expected results. SPF, a pure link-state protocol that has not been optimised or modified for wireless ad hoc routing, is the most expensive protocol in all scenarios. EXBF and DSDV, two distance-vector protocols, have a similar performance profile with DSDV edging out EXBF. AODV and DSR perform well, especially for lower traffic loads. TORA exhibits the poorest performance; this may be attributed to its lengthy route discovery process.

In terms of packet delivery rates, both the proactive link-state and distance-vector protocols perform well. AODV and DSR perform poorly in comparison, dropping many packets during the route discovery phase. This observation is specific to this study as send buffers are an integral part of both protocols in their fully implemented form. The absence of early route request quenching in AODV and DSR leads to longer delays in the route discovery phase, causing more packets to be dropped. Allowing for these aberrations from the intended operation of AODV and DSR, AODV still performs marginally worse than DSR. TORA has a similar delivery rate performance even though it is a multi-path protocol. TORA is also more adversely affected by routing packet loss.

SPF, EXBF and DSDV show the lowest average end-to-end delays for all tested scenarios. AODV and DSR do not perform as well as their routes are generally not the shortest. AODV's routes are maintained in a form of soft-state, timing out after an interval to be refreshed by a new route discovery phase. This means that new, shorter routes may be discovered after a time, allowing it to edge out an advantage over DSR's performance. TORA has the worst delay characteristic.

To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV, DSR**: The incomplete implementations of these protocols yield results in this study that may favour the other protocols. The protocols exhibit atypical behaviour when compared to results from other studies.
- **DSDV, EXBF, SPF**: These protocols have excellent end-to-end delay and delivery rate performances. However they all have a considerably higher control overhead cost which cannot be throttled back when traffic loading is low.
- **TORA**: Performs poorly overall. The cost of maintaining multiple paths and its sensitivity to routing packet loss outweigh the benefits of having access to multiple-paths in even moderately mobile networks.

VII. A study of the AODV, DSR and OLSR protocols is presented in [Cla02b] which used the ns-2 platform for performance evaluation. The authors of this study are the designers and active proponents of the OLSR protocol. The scenarios evaluated were characterised by the use of 50 nodes in a simulation field of 1000m by 1000m. The node density was kept constant at 1 node per 20,000m^2. The transmission coverage area for each node was 196350m^2, yielding an average node degree of 9.8. The RWP model was used to determine mobility with node speeds of 1-5 m/s and pause times of 0-5s used. The traffic load in the simulations was provided by 25 CBR streams sending packets at a rate of 10 packets/s in bursts of 10s. The authors also evaluated the performance of the three protocols under TCP traffic, varying the traffic from 6 to 20 streams. This type of traffic is rarely used in these studies for the reasons outlined in Section 2.4.3, concerning the conformal nature of TCP. The authors primarily focused on packet delivery, end-to-end delay and control overhead as the key indicators of the performance of each protocol.

The packet delivery performances of the protocols indicate that the two reactive protocols, AODV and DSR, have a similar performance profile while AODV is slightly better in more mobile networks. DSR and AODV

also outperform OLSR in high mobility scenarios as they buffer packets, something that this implementation of OLSR does not incorporate, in agreement with the design specifications. However, for moderately high levels of traffic, OLSR outperforms both of the reactive protocols. The higher levels of traffic trigger the maintenance and discovery mechanisms of AODV and DSR more often, leading to more congestion with higher control overhead for these two protocols. The control overhead of OLSR has no correlation to traffic levels.

The end-to-end delay characteristics determined by this study follow those of other studies. AODV and DSR are penalised for buffering packets during the route discovery phase, whereas OLSR drops packets for which it does not have a valid route.

Not surprisingly, the overhead of OLSR is constant irrespective of changing node mobility or traffic loads. OLSR's control overhead is less than one seventh that of DSR and one tenth of AODV at the highest level of mobility. In this study, both AODV and DSR are found to incur greater control overhead with increasing node mobility, with AODV being twice as expensive as DSR at the highest mobility rates. When the protocols were stressed by an increased number of CBR traffic sources, the performance of AODV deteriorated continually, showing a marked increase in control overhead after only a moderate increase in CBR traffic. However, the control overhead incurred by DSR, while increasing slightly under a moderate increase in CBR traffic load, plateaus off under continuing CBR traffic load increases. At the highest CBR traffic loading AODV's control overhead is three times as expensive as that of DSR.

Finally, each of the protocols is evaluated under a range of TCP streams. However, due to the conformal nature of the TCP scheme, the value of the evaluations may be questioned. The load offered by TCP does not stress-test the protocols as it reacts to dropped packets by lowering the data rate. TCP assumes that dropped packets are due to congestion whereas they are more likely due to broken links in a mobile ad hoc network. OLSR shows the best performance, having the advantage that paths are set-up in both directions for the bidirectional TCP stream. The authors note that TCP is used on about 95% of the existing Internet, but as has been noted in Section 2.2, the mobile ad hoc networking environment demands are quite different from the conventional stable wired Internet, demanding new approaches to flow control. Therefore, the results yielded by the application of TCP traffic loads, while interesting, are not as good an indicator of performance efficiency as those gained from the application of CBR-loads.

The authors conclude that none of the protocols is suited to all networking conditions, while each of the protocols has strengths under particular conditions. To summarise, the following observations may be made about each of its candidate protocols based on the information provided in the study:

- **AODV, DSR**: The buffering provided by both AODV and DSR give them an advantage over OLSR in highly mobile situations. In lower mobility situation DSR outperforms AODV in delivery terms. Both of the reactive protocols suffer from higher overhead with increasing levels of data traffic; their delivery performance degrades due to increased congestion. The end-to-end delay performance of these protocols it not as good as that for OLSR. This is caused by two factors. Firstly, the protocols do not always use the shortest path, and secondly, the protocols buffer packets while discovering routes. The average path length supplied by AODV is found to be longer than that used

by DSR. The automatic route shortening mechanism used by DSR may help it to improve routes as it may eavesdrop on all passing routing information. In very active networks, the control overhead generated by these protocols is generally higher than that of an OLSR network for the same mobility conditions.

- **OLSR**: In high mobility scenarios, OLSR is penalised by the absence of a data buffer. However its delivery rate exceeds AODV and is roughly equivalent to DSR at lower mobility rates. OLSR's control overhead is independent of traffic rates and has a better delivery performance than the reactive protocols at higher traffic rates. OLSR's end-to-end delay profile is helped by the fact that it uses the shortest path and it does not buffer packets for which it does not currently have a route. OLSR suffers in quiet networks as it cannot throttle back on its control overhead when there is no demand.

VIII. Another study of the AODV, DSR and OLSR protocols is presented in [Vie03] along with a proposed framework that allows for the probabilistic modelling of network topology and data traffic. Again, the authors of this study are the designers and proponents of the OLSR protocol. This study differs from the other cited evaluations quite significantly. The networking environment for this study was characterised by the creation of a probabilistic framework that allowed for the modelling of reactive and proactive protocols, network topologies and data traffic. The objective of this particular evaluation study was to develop a model that would allow for the analysis of control traffic overhead in mobile ad hoc networks in order to identify which protocol is better suited to a particular situation. The main result that this paper presented is that there is a reasonably simple model for the relationship between node mobility and control overhead for both classifications of protocols. The main parameters that are varied in this experiment are the node mobility, modelled by changing the link failure rate, and the number of active routes, which models the traffic activity in then network.

As part of an exercise to validate the model, the authors then compared their results for AODV and DSR to those obtained in three other papers [Joh99], [Bro98] and [Das01]. The results obtained in this evaluation agreed with results obtained from similar experiments described in [Joh99]; in particular it was noted that the number of link changes is approximately proportional to the mobility metric used in that study. The results in [Joh99] indicated that there is a close, near linear, relationship between the increase in the emission of control packets and increasing node mobility for both AODV and DSR.

Upon comparing their results with the [Bro98] work, the authors have again found that an approximately linear relationship exists between increasing control overhead and increasing node mobility for both AODV and DSR. The results from the [Bro98] paper agreed with the vast majority of the results from similar experiments in this paper [Vie03]. The results obtained for experiments on the OLSR protocol are unsurprising. There is found to be no correlation between the control overhead and traffic loading.

2.4.5 Summary of Published Evaluations

The samples of evaluation studies that have been reviewed in the previous section illustrate the diverse capabilities, efficiencies and inefficiencies of many ad hoc routing protocols. The many scenarios that were

presented in those evaluations stress the candidate protocols. The following summaries of observations have been taken from the published evaluations to illustrate the different levels of performance that may be expected of these ad hoc routing protocols.

- **AODV** has been evaluated in [Bro98], [Joh99], [Per02], [Das01], [Das98], [Cla02b], [Vie03] and [Cla02c]. It is evident from the frequency with which AODV features in the selection of evaluation papers presented in this chapter that AODV has proven to be a popular and well-established protocol against which other protocols can been be benchmarked. The amount of control overhead incurred by AODV is proportional to the mobility of the nodes. AODV only incurs control overhead during the route discovery and route maintenance phases. It does not incur any overhead when forwarding data. AODV also performs well when offered a destination-diverse traffic loading. High node mobility has a detrimental effect on the throughput and delay performance of AODV.

- **DSDV** has been evaluated in both [Bro98] and [Das98]. Both of these studies indicate that DSDV has trouble converging when the mobility is increased to a moderate level as it is an event-triggered protocol, i.e. a protocol that issues control packets when routes fail. The protocol generates too much control data which congests the network in such situations. The protocol does not adapt to mobility or traffic loading changes, so that in quiet and stable networks DSDV's control overhead is relatively high.

- **DSR** has been evaluated in [Bro98], [Joh99], [Per02], [Das98], [Das01], [Raj00], [Cla02b], [Vie03], [Raj01] and [Lee99]. It is again evident from the frequency with which DSR features in the selection of evaluation papers presented in this chapter that DSR, as with AODV, has been a popular and well-established protocol against which researchers have chosen to benchmark other protocols. The throughput performance of DSR is penalised when the network is offered destination-diverse traffic loads; its use of source routes has a congestive effect on the network.

- **OLSR** has been evaluated in [Cla02b], [Vie03], [Cla02c], [Lao02] and [Cla01]. The evaluations of OLSR have demonstrated a number of expected properties regarding the protocol. The control overhead incurred by the protocol is independent of traffic and almost entirely independent of mobility; some implementations issue aperiodic Hello messages when critical relay neighbours are lost due to relative node mobility. OLSR's lack of buffering makes the delay figures look impressive as it just drops packets. OLSR's strength is in dense, lower mobility networks in which there are many communicating pairs of nodes, i.e. a destination-diverse traffic loading. In higher mobility situations OLSR is slower to converge and it has a degraded delivery performance.

- **TORA** has been evaluated in [Bro98] and [Das98]. These evaluations of TORA have not been favourable to the reactive distributed multi-path protocol. As it is reactive, the control overhead that it generates is correlated to both the mobility and traffic loading of the network. The [Bro98] study, which was a faithful implementation of the protocol, reported that it had an excellent packet delivery profile when the traffic load was low, less than 20 CBR streams in a 50 node population. Higher levels of traffic led to congestion in the network as TORA generates a lot of control packets in its lengthy route discovery phase. Increased mobility also affects the control overhead as TORA relies on IMEP to provide reliable and ordered communication with its neighbours and to proactively sense the neighbourhood.

These non-hierarchical protocols represent five different approaches to multi-hop ad hoc routing. They range from the reactive and distributed distance-vector protocol of AODV to the improved proactive link-state protocol of OLSR. The evaluation studies have shown that the protocols have weaknesses and strengths with regard to their performances under different networking conditions. It is abundantly clear that no single protocol is optimal in all networking conditions. It has been noted in many of the above-cited studies that, in addition to the ad hoc routing protocol in use, the other significant factors affecting the performance of an ad hoc network are the relative mobility of the network's nodes, the degree of one-hop connectivity of those nodes and the type of traffic that is loaded onto the network. In terms of the evaluations, these networking conditions are determined by the values of the node mobility, node density and traffic loading parameters of the simulator engines.

As an example, the study presented in [Cla02b] demonstrates that DSR and OLSR have very different performance profiles when tested under conditions of low mobility (i.e. 0-5 m/s), high density (i.e. about 10 neighbours per node) and moderately high traffic loading (i.e. more than 25 CBR streams in a population of 50 nodes). OLSR nodes achieve 50% more throughput than DSR nodes. However, when the conditions are changed to reflect a higher mobility environment, i.e. node velocities greater than 5 m/s, then the throughput performance of OLSR degrades significantly. Under such networking conditions, the throughput of OLSR falls off in comparison to that achieved by DSR; OLSR only delivers 50% to 60% of the packets that DSR manages to deliver.

While these networking conditions have influence over the network-layer, the network-layer generally cannot exert any influence over these conditions, i.e. the network-layer cannot change the conditions to suit its demands. The actual values of these factors, i.e. the number of neighbours per node or the relative mobility of two neighbouring nodes, are outside the scope of the network-layer's control, given the type of ad hoc routing protocols described in the studies. Some power-aware protocols have been designed for particular radio front-ends which enable the node to control and vary the transmission power of the radio and therefore the degree of the node. These protocols have the aim of minimising power usage by maintaining network connectivity using the minimum power possible. However, the routing protocols that were evaluated do not exercise any control over the radio's transmission power and transmission radius, thereby lacking the ability to change the degree of the node. Furthermore, the autonomous movement and positioning of mobile nodes, which has a direct bearing on nodes' relative mobilities, is not dictated by the network-layer. Also, the type and volume of traffic that is loaded onto the network is decided by higher layers such as the application or

transport layers; the network-layer simply provides a data forwarding service based on the fundamental network building block of the unreliable data packet. Essentially, the conditions faced by network-layer protocols are determined by external forces and events.

With a view to developing a networking system that leverages these existing and somewhat matured ad hoc routing protocols, it is insightful to consider and understand the manner in which the above-mentioned networking conditions actually affect the operation of the ad hoc routing protocols. The routing protocols are run locally at nodes as part of a distributed network-wide system; each node takes local actions and makes local decisions in accordance with the chosen protocol. The efficiency with which each node can execute the various elements of the routing protocol, e.g. the route discovery mechanisms, the route maintenance mechanisms or the neighbourhood discovery mechanisms, affects the performance of the network, e.g. the average throughput achieved or the average control overhead incurred.

■ **Effects of Node Mobility**

Fundamentally, the performance of a mobile ad hoc network is related to the efficiency of the routing protocol in adapting to changes in the network topology due to the mobility of nodes. As described in Section 2.4.2, the mobility of nodes in simulations is determined by mobility models such as the RWP model which represents mobility in terms of minimum and maximum velocities and pause times. In much of the literature the pause time is used to describe the mobility of the nodes, i.e. a scenario in which the pause time is 0s is considered to be highly mobile, i.e. the nodes are constantly moving, whereas one in which the pause time is 120s is considered to be fairly static, i.e. the nodes are stationary for relatively long periods.

Some of the literature has inferred a causal relationship between the relative mobility of the nodes and the performance of the protocol, [Joh99], [Kwa03], and [Vie03]. This is an aspect of node mobility that the pause time, minimum velocity and maximum velocity parameters of mobility models such as the RWP or Nomadic Community models do not describe or measure. Indeed, the evaluation study presented by [Per02] has shown that there is no direct correlation between pause times per se and the performance of a routing protocol. Notwithstanding that observation, mobility models such as the RWP model are suitable for testing the effects of relative node mobility on the ad hoc routing protocol's performance as they emphasise the local effects of relative mobility. There is no association between nodes in the RWP model, i.e. the movement of one node is not tied to, or influenced by, the movement of another node, and consequently nodes are independently assigned velocities leading to *randomised* scenarios that involve many link changes. Group Mobility models or Nomadic Community models do not place the same stresses on the routing protocol as the relative mobility of the nodes within a group is purposefully insignificant when compared to the relative mobility of various groups or communities, thus placing less stress on node-to-node links.

The stability of links between neighbouring nodes is directly related to their relative mobility which influences the rate at which links come up or go down, i.e. the rate of one-hop link changes. It has been shown in both [Joh99] and [Vie03] that there is an approximately directly proportional relationship between the rate of link changes and the increasing relative mobility of the nodes. When nodes move in and out of range of each other, new one-hop links may be detected and existing one-hop links may fail. The manner in which a node detects, and deals with, such link changes has a bearing on the performance of the protocol.

Both the relative mobility of nodes, as manifested in the rate of change of one-hop links, and traffic loaded on the network are the two stimulants that directly affect the amount of control overhead incurred by reactive protocols. Control overhead is costly as there is generally a per-packet cost in accessing the wireless medium, leading to contention delays and collisions which diminish the available bandwidth. For this reason, the minimisation of the amount of control overhead incurred is an important objective of reactive routing protocols. Mobility only affects nodes using reactive protocols if they are supporting traffic streams; in the absence of traffic, nodes operating reactive protocols do nothing. The effects of different types of traffic loading are discussed later.

Generally, a reactive protocol discovers that an actively used link has gone down when it observes a failure event such as a data packet not being acknowledged by the next-hop node. AODV also uses proactive neighbour sensing on active routes so that it can detect failed routes faster and possibly correct the fault before a data packet is to be routed over the broken link. However, if the rate of link failures increases due to increased node mobility then the proactive neighbour sensing mechanism may not be sufficient to pick up on link changes in a timely manner. Shortening the interval at which the neighbour-sensing packets are sent may solve the problem, but it incurs extra overhead. AODV attempts a localised link repair process before proceeding to report the error to the source node which may initiate a fresh route discovery process. Local link repair is used to limit the effects of a link failure; essentially a route discovery request of limited scope is sent out to query local nodes' route caches. The effectiveness of AODV's local link repair process is limited by the fact that nodes do not cache overheard routing information and they do not maintain multiple route entries for any destination node.

In the case of DSR, a link on an active route may break while the route is not in use but the protocol will not attempt to repair the link until the nodes using the broken link attempt to use it again. DSR nodes may try to locally salvage the data intending to use the broken link by checking their route caches for alternate routes to the destination, a process that incurs no overhead. Unlike AODV, DSR nodes can store multiple routes to the same destination which they may have learned from multiple route replies or by eavesdropping on control packets that are issued by neighbouring nodes. DSR's caching policy is more effective at lower speeds when routing information goes stale more slowly. Otherwise, DSR nodes report the broken link to the source node which is responsible for initiating a fresh route discovery process.

As a result of these route maintenance procedures, it is observed in [Bro98], [Das01] and [Joh99], that there is a near linear relationship between the control overhead incurred by both AODV and DSR in the presence of increased node mobility. Both protocols follow very similar curves, although there is a significant absolute difference between the two protocols with AODV consistently incurring more overhead in terms of packets. However, the effect of node mobility on the amount of control overhead is strongly correlated to the traffic loaded onto the network, as is discussed later. In a largely static network, with few link breaks, reactive protocols such as AODV and DSR incur very little control overhead and have a high throughput performance as there is very little need for route maintenance. This is the only situation in which they incur less overhead than a proactive protocol such as OLSR.

The control overhead incurred by largely time-triggered proactive protocols such as OLSR is almost immune to node mobility as such protocols do not explicitly react to all link failures. While mobility degrades the throughput performance of the protocol in the absence of an increase in node density, it does not cause the type of congestion caused in reactive networks. OLSR does incorporate an event-triggered mechanism which allows nodes to locally recompute routing information when certain *important* links fail, however this process is localised and has not been found to have any significant effect on the overall overhead incurred [Vie03]. Overall, while increased node mobility does not affect the control incurred by OLSR, it does affect the throughput as the network does not converge as fast. Unlike reactive protocols, OLSR does not throttle its response to node mobility and in moderate to highly mobile scenarios it generally has a lower average throughput than either AODV or DSR as they can react to increased mobility. The end-to-end delay performance of OLSR does not suffer due to increased mobility as OLSR drops packets for which it does not have a route. OLSR only delivers packets if the route is immediately available whereas AODV and DSR will buffer them while attempting to discover a route. Consequently, the end-to-end delay performance of AODV and DSR increases with increased mobility.

Finally, while the use of link-layer feedback from MAC protocols such as the IEEE 802.11 DCF MAC may offset some of the local negative aspects of increased node mobility by providing timely link status information to the network-layer, the literature has shown that such MAC protocols have a negative impact on the global throughput of ad hoc networks [Li01], [Xu01]. Also, the use of such proactive link sensing at the MAC-layer to facilitate the operation of the network-layer defeats the purpose of designing a reactive network-layer component, as AODV and DSR have been designed. Indeed, the proactive OLSR has been specifically designed to be independent of the link-layer [Cla02b]. Furthermore, the use of MAC-layer link sensing introduces cross-layer dependencies which inhibit the design of flexible protocol stacks. In summary:

I Node-to-node link breaks require protocols to discover/maintain routes efficiently. Above all else, protocol performance depends on the ability to respond quickly to such changes in the topology.

II RWP is a tough mobility model as it poses the most robust challenge to node-to-node links.

III There is a directly proportional relationship between relative node mobility and the rate of node-to-node link changes brought about by that relative node mobility.

IV For reactive protocols, control overhead is proportional to the number of traffic streams being supported and the stability of the node-to-node links over which those streams are being routed.

V For time-triggered proactive protocols, while the amount of overhead incurred is independent of the traffic loaded onto the network, such protocols cannot react to increased mobility by incurring more overhead.

■ Effects of Node Density or Degree

As described in Section 2.4.2, the network size (node population), the transmission radii of the nodes and the physical simulation area are the simulator parameters that dictate both the global scale of the network and the local node density. Protocol designers often examine the effects that increasing the network size will have on the protocol, i.e. they investigate the scalability of the protocol. In general, a larger population of nodes dispersed over a larger area yields a greater network diameter. Large network diameters stress the scalability

of a network protocol by demanding efficient (i.e. fast and cheap) route discovery procedures in reactive protocols and quickly converging topology information diffusion algorithms in proactive protocols.

However, it was noted in the evaluation studies that node density has a more pronounced bearing on the performance of the routing protocols than the network diameter; with node density having a more immediate and localised affect on the network performance. The node degree of each node is directly related to the number of nodes in direct transmission range of a given node, describing the local possible one-hop connectivity of a node. Such one-hop connectivity is the basis upon which the multi-hop connectivity of ad hoc routing protocols is built. If a wireless node uses less power the transmission radius is shorter and the number of collisions should be less, since there are less overlapping transmissions at the receiver. If the transmission radius is increased the packets go further and the possibility of discovering nodes in the *right* direction is enhanced, but the number of collisions also increases which reduces the effective bandwidth.

The specific effects of a node's neighbourhood size on the performance of wireless networks has been studied in [Nil04], [Roy01b] and [Tak84] among others. Takagi and Kleinrock [Tak84], updating and correcting their more popularly cited paper of 1978 [Kle78], argued that the optimal node degree for nodes in a stationary packet-radio network was between 7 and 8; the earlier paper had suggested 6 to be the so-called *magic* number. Takagi and Kleinrock investigated the optimum transmission radii for randomly distributed nodes (i.e. using a 2-D Poisson distribution) in a stationary packet-radio network, the precursor networks to wireless mobile ad hoc networks, in an effort to discover if there was an optimum number for the node degree in packet radio networks using slotted ALOHA. The authors arrived at this result by considering the optimal transmission radii necessary for the maximisation of the expected one-hop progress of a packet in a desired direction. The authors drew a hypothetical line from the source node to a destination node and the transmission radius for each node was chosen such that the expected distance advanced in one transmission, projected onto the line, was maximised. In the literature on the optimisation of transmission radii this approach is referred to as *maximised forward progress* routing.

Royer *et al.* [Roy01b] have looked at the effects of transmission radii on the performance of mobile ad hoc networks, specifically examining the effect that mobility has on the optimum transmission radii for nodes in ad hoc networks. Using simulations of AODV, the authors found that the *magic* number for AODV in a static network was 7 or 8 neighbours per node, concurring with the results of Takagi and Kleinrock [Tak84] whilst using a completely different methodology to arrive at this result; the authors' aim was to maximise the throughput performance of a simulated AODV network. When Royer *at al.* investigated the effects of mobility they found that there was no global optimum that suited all node mobilities; rather, they determined that as node mobility increases the optimum number of neighbours also increases.

As noted in the last section, when nodes move faster relative to each other, link changes occur more frequently. To compensate for the increased instability of links it proves beneficial for a node to have more neighbours, i.e. a higher node degree, so that it can use alternate links to repair a route. In the [Roy01b] study, increased node degree was brought about by increasing the transmission radius of the node which results in increased collisions and reduced bandwidth. However, when compared to scenarios having a lower node degree and equally high mobility, the increased transmission range yields a higher throughput as link

breaks are less frequent and routes are maintained for longer periods of time. An increased transmission radius means that two nodes will remain within communication range for longer, thus reducing the rate at which link changes occur. However, there is a limit as to how far the transmission radius can be increased as the network becomes saturated beyond a certain point; the number of collisions increases and increased channel contention decreases the effective bandwidth considerably. While increased node density has been shown to compensate for increased node mobility for AODV, in low mobility situations it has been shown to lead to network saturation. Royer *et el.* also showed that at very low densities the network does not stay connected; many nodes or groups of nodes become disconnected from the main network partition and traffic sessions are aborted.

Indeed, the absolute difference between the control overhead incurred by AODV and DSR decreases with increasing node density as there is more relaying of flooded route requests for AODV. DSR nodes send less route requests in more dense environments; it has also been noted in [Mal99b] that increased node density enables DSR nodes to overhear more routing information. As DSR control packets containing source routes traverse the network, eavesdropping neighbouring nodes may cache the routing information. This routing information can subsequently be used to answer low-cost non-propagating route requests from neighbouring nodes, thereby cutting down on the number of fully propagated requests that incur a much larger cost on the network.

OLSR is purposefully designed for high density scenarios as it uses an optimising technique to minimise the number of neighbouring nodes that relay broadcast messages for the originating node; a technique that becomes more efficient as the node density increases. Thus, increasing density is ideally suited to OLSR for which the main negative factors are node mobility and traffic. In summary:

I Too much node connectivity leads to higher contention for the wireless medium and a consequent loss of effective bandwidth.

II Too little node connectivity increases a node's dependency on the stability of its existing neighbours.

II As a node's mobility relative to other nodes increases, its optimum node degree also increases as the increasing instability of its existing neighbours causes the node to seek alternative routes.

III For reactive protocols, the caching of routes based on eavesdropped information is beneficial in higher density environments as it reduces the need to explicitly query fresh routes.

IV The use of an optimised set of neighbouring relay nodes for diffusing broadcast packets becomes more efficient with increasing node density.

■ Effects of Traffic Loading

It is not surprising that traffic is a significant factor in the operation of certain types of ad hoc networks. Although the primary purpose of an ad hoc network is to facilitate multi-hop peer-to-peer communication, i.e. to support traffic, ad hoc routing protocols vary in their ability to cope with different types of traffic. As noted in Section 2.4.2, the traffic that is loaded onto an ad hoc network is characterised by the number of destinations that are being targeted with data, the number of concurrent streams and the volume of those streams.

On-demand protocols such as AODV and DSR only react to multi-hop communication needs which are fuelled by traffic loaded on to the network. Therefore, for reactive protocols, all of the points noted with regard to the effects of node mobility on the performance of reactive protocols are only pertinent in the presence of traffic. The nature of that traffic may amplify the effects of node mobility. When there is demand for routes to multiple diverse destinations by various source nodes, the number of route discovery procedures initiated by nodes increases. Individual nodes may limit the rate at which they process or initiate requests. AODV nodes throttle the rate at which they generate route requests, while both AODV and DSR use a binary exponential back-off algorithm coupled with expanding-ring search techniques to control the rate at which they re-issue route requests.

Increased numbers of destinations also have implications for the route maintenance phases of reactive protocols. If a node is supporting multiple streams it is responsible for maintaining the routes if they breakdown on an outgoing link. If links around a reactive node are unstable it may have to engage the route maintenance procedures more often. In this regard, traffic focused on a few destinations does not place much stress on reactive protocols. Long-lived routes are also preferable in a reactive environment whereas proactive protocols such as OLSR are more efficient in the face of scattered traffic patterns as routing information is constantly available for all nodes connected to the network. Using OLSR, there is no additional overhead required to support requests for multiple destinations or to support data en route to multiple destinations, no matter how long or short the end-to-end connection is. On the downside, OLSR cannot throttle back its overhead when there is no traffic loaded onto the network.

The evaluation studies revealed that OLSR has a higher throughput than AODV or DSR as traffic loading increases in lower mobility environments. This is mainly due to the stable overhead costs of OLSR with regard to increasing traffic. However, as mobility increases, it becomes a more dominant factor than the traffic loading and the performance of OLSR degrades so that AODV and DSR have a better throughput profile. With increasing traffic in moderate to highly mobile situations, AODV outperforms DSR in terms of throughput as traffic loading increases.

Nilson [Nil04] has examined the combined effects of node degree, node mobility and traffic loading on the performance of AODV and OLSR networks. The author has shown that the optimum node density is dependent on the traffic loading and the node mobility. Not surprisingly, it was found that smaller node densities result in lower average network connectivity which in turn leads to less routes being found or created and, consequently, less throughput across the network. In concurrence with [Roy01b], it was found that node density must rise as mobility rises to maintain the same throughput performance for AODV.

In the case of AODV, Nilson found that low node densities (i.e. less than 5 neighbours per node) lead to lower network connectivity and that consequently the average throughput is lower due to a lack of routes. However, when the mobility increases the connectivity rises the throughput also increases for small traffic loads. As the mobility increases, the optimum number of neighbours also increases. Nilson also found that as traffic loading was increased in networks experiencing better connectivity, the throughput fell for AODV as it led to more congestion. In sparse AODV networks, when the network manages to establish a route for the loaded traffic it experiences better throughput as there is less channel contention. In concurrence with the

earlier evaluation studies, OLSR was found to yield a high throughput rate in dense networks. The increased node density leads to increased network connectivity and the discovery of more routes. Also, as already noted, increased network density allows the message relaying minimisation feature of OLSR to become more pronounced. As OLSR copes better with dense networks it is more likely to have routes than the reactive protocols; consequently it can deliver a higher average throughput. In summary:

I Proactive protocols continue to incur overhead in the absence of traffic, whereas reactive protocols are costless when there is no traffic.

II Reactive protocols perform well when nodes are requested to create long-lived routes to a small number of destinations. Increasing traffic destination diversity amplifies the negative effects of node mobility and node density for reactive protocols.

III Proactive protocols, having access to total topology information at all times, are efficient when presented with traffic destined for multiple destinations, regardless of the duration of the traffic streams.

■ **Adaptive Local Tuning**

Some ad hoc routing protocols have local parameters which may be varied in order to improve the performance of a node; however, there is a limit to the extent to which the variation of these parameters enables the protocols to act in a more efficient manner. It should be noted that the nature and function of such local adaptation is outside the scope of the protocols' specification documents. Reactive protocols such as AODV and DSR *could* have variable control over the timeouts which apply to their route cache entries and the rate at which, and extent to which, they search for fresh routes. The variation of the route cache timeout parameter may enable the nodes to expunge route cache entries before they become stale; a node may decide to drop routes more quickly in more mobile situations whereas a node may assign a longer life-span to routes in lower mobility networks. Nodes could expunge route entries using this technique in an attempt to pre-empt a normal route maintenance phase. Routes are normally expunged when they are found to be broken during the process of packet-switching or when their statically configured timeout expires. By pre-empting the break-down of a route the protocol may save time wasted on discovering the break, preferring instead to discover a fresh route during the time that would otherwise have elapsed in making that discovery. The other parameter over which protocols may exercise control is the type of search technique that they use; changing from expanding ring searches to non-propagating searches and varying the timeouts between consecutive search attempts.

If nodes are enabled to unilaterally change such variables as the rate and extent to which they query routes to destinations, then the network will experience the occurrence of greedy nodes which increase control overhead incurred on the network to their own end. Such unilateral action may have negative effects in other areas of the network which may experience increased congestion and medium contention.

Furthermore, it has been noted that one of the most negative factors for DSR is the occurrence of demands for routes to multiple destinations; DSR cannot exercise any control over this type of traffic loading and an ability to tune its parameters would be of no use in such a situation. So, while the adaptive tuning of such local protocol parameters in response to changing network conditions allows the nodes to micro-manage the situation, it does not enable the protocols to overcome inherent incompatibilities between their design and the

networking conditions that they experience. When the conditions experienced by nodes are significantly different to those that they are suited to, then these local adaptations do not suffice.

■ **Concluding Remarks**

Finally, it is worth noting that the cited evaluations, in common with most existing research, have focused on the deployment of ad hoc networks in closed environments. Standardised evaluation trials were used in [Bro98], [Cla02b], [Das01], [Das98], [Joh99], [Per02] and [Raj00] to simulate closed networking scenarios where uniform networking conditions were experienced throughout the simulation space; the node mobility, node density and traffic loading values were not varied for the duration of the experiments. Indeed, in the scenario-based trials of [Joh99] the same conditions were imposed across the simulation space for the entire duration of the evaluations.

Whilst this approach to ad hoc protocol evaluation has obvious merits, in that it allows for the comparison and study of various protocols under specific networking conditions, it tends to give a misleading impression of the environments in which real ad hoc networks will operate. Real-world ad hoc networking environments cannot be modelled by a single invariant simulation space such as the standardised and scenario-based simulation spaces used in the aforementioned evaluation studies. Real ad hoc networks will exist in more challenging, open, dynamic and diverse environments. In order to assess the needs of an ad hoc networking system that will cope such networking environments the next section explores contexts in which ad hoc networks will most likely feature in the future.

2.5 Ad Hoc Networks for the Future

Research into next-generation wireless networks foresees the adoption of multiple and evolving standards. Ad hoc networks have been seen by some as a possible next step in the evolution of so-called second and third generation systems [Fro01]. The authors envisage a future generation of wireless networks that utilise multiple air interfaces; Bluetooth, IEEE 802.11, WCDMA, EDGE and cdma2000. The necessity of future networks to interoperate with current standards is certain given the enormous sums of capital invested in infrastructure in the last decade. It is suggested that ad hoc networks may complement other infrastructure-based 4G networks. Ad hoc networking is viewed as a technology to further the concept of always-best-connected in future generations of communications systems.

While [Fro01] envisages a flexible always-best-connected concept built around existing architectures, alternative views of future network architectures have also been promoted in [Bra00] and [Cas96]. The evolution of the conventional Internet has led researchers to reassess the requirements of future network systems that may have to cope with a more diverse range of constraints and opportunities, whether they are technological, economic or political. The development of ad hoc networking solutions and standards should not take place in a vacuum, ignorant of the parallel advances and proposals in the field of wired Internet research.

[Bra00] discusses many areas of interest in next-generation architectures, highlighting the factors that should motivate the design of such systems. Among the requirements underlying the original Internet architecture

were internetworking, robustness, heterogeneity and distributed management. While these objectives were met in the design of the original Internet, within the context of its existing stable and wired form, the solutions are no longer adequate and will not suffice into the future. A network architecture is expected to be relatively long-lived, applicable to more than one generation of a technology, e.g. IPv4 [Uni81] and IPv6 [Hin98] use two different addressing schemes that both conform to the same Internet architecture. The role of such an architecture is to ensure that the resulting technical designs of various schemes, such as routing protocols and addressing schemes, will fit together smoothly. The requirements of the Internet and other networks that extend or complement it, such as ad hoc networks, continue to change and become more demanding. [Bra00] suggests that some of the new requirements that should influence new designs are mobility, auto-configuration and time-varying resources, among others. The author suggests that a new architecture should be flexible and should efficiently support highly dynamic mobility. Current ad hoc routing protocols are an example of schemes that fulfil that objective, but they do so within the inflexible framework of the existing Internet structures.

The authors conclude that the long-term viability of the Internet requires a more revolutionary approach to the existing framework, although they do not propose a possible solution. They argue that any new design should be independent of short-range issues and should not require backward compatibility. The ability of ad hoc networks to exist as standalone networks will enable such experimentation to take place without having to take account of existing standards and practices.

The Nimrod [Cas96] architecture is a scalable routing architecture that has been designed to accommodate a continually expanding and diversifying internetwork. The authors have targeted Nimrod for an internetworking environment that is unpredictable. They suggest that a range of political alliances, technological advances and demand economics will determine how such an internetwork will change over time. Therefore, the authors' overriding concern is that the architecture is flexible enough to incorporate unanticipated changes to any of the environmental constraints that underlie the network's operations. Such constraints may include network elements such as routers and hosts. They may also include the number of hosts and routers in the network. Other constraints include specialised user requirements, e.g. low latency streaming, and traffic session dynamics.

The future of the mobile ad hoc networking environment is one which will most likely see a fairly rapid pace of turnover in its constituent elements. The ad hoc networking system may be very heterogeneous. There are currently a variety of services available for each layer of the communication stack. This variety already poses problems in the accurate and representative simulation of ad hoc networks. As ad hoc networks may be developed as standalone networks, it is open to developers, vendors and other authorities to adopt protocols and solutions from a variety of available proposed standards. Such decisions may be grounded in technical, political or financial reasoning. The components of ad hoc networks may be upgraded at a faster pace than was previously possible in large interconnected hierarchical systems. Already, the IEEE 802.11 standard, which has been a popular initial choice for wireless transmission in ad hoc networks, has undergone a number of revisions and improvements in a relatively short period of time. Ad hoc networking systems need to be flexible enough to accommodate change in the short to medium term.

2.6 Ad Hoc Networks in an Urban Context

Much of the research into ad hoc networking technologies has investigated them in closed and controlled environments. The evaluations that have been reviewed were typical of that approach to the design of ad hoc networking solutions. The studies investigated the use of ad hoc routing protocols in closed environments with uniform networking characteristics. Closed, controlled environments range from military scenarios such as battalions in the theatre of war to automated, self-auditing machinery on a factory floor and networking delegates in a conference venue. In closed and controlled environments the configuration of the nodes participating in the network can be tightly managed to ensure that the choice of ad hoc routing protocol suits the networking dynamics of the situation. Nodes are generally assigned to a common task, configured prior to deployment and are of a common origin. The introduction of foreign bodies into such a system is very unlikely, if not technically impossible.

The real challenges for ad hoc networks are not those that exist in closed environments where the inherent qualities of the network are often stifled by artificial constraints such as human authority structures or the immobility of a factories plant. Rather, the real challenges in ad hoc networking development come from more open environments. An open ad hoc network is one that exists without any sort of pre-existing supervising infrastructure or authorities such as a military commands, college authorities or system administrators. An open distributed network allows any node to join or leave the network subject to the meeting of certain technical and security criteria. Among these criteria are the ability to use a common networking protocol, the ability to develop a trust based on the sharing of credentials and the ability to trade services in a cooperative and beneficial manner.

It should be noted that the *real world* is not a tightly controlled environment. As people are neither machines on the factory floor nor, generally, soldiers on the battlefield, the ad hoc networks that may be used by civilian populations will be found in a range of diverse locations and scenarios. Such open networks will occur in situations where there exists a body of users and devices coupled with opportunities to exploit interesting or necessary applications and services. The fact that ad hoc networks may 'happen' or 'occur' is one of the traits that most strikingly differentiates them from other networking and communications standards. One such environment that will support the creation of open ad hoc networks is the urban environment. The rest of this section seeks to explore the relationship between ad hoc networks and the urban environment. By examining past experiences and observations of the relationship between urban life and telecommunication systems and by forecasting the possible interaction of ad hoc networking technology with contemporary urban life, the diverse, motivating context for this thesis is presented.

The civilian deployment, adoption and use of ad hoc networks may be a complex task. Civilian users, unlike military users or automated machines, exercise choice in their adoption and use of new technologies. The rate of adoption of new technologies and the variation in the use of the technologies adds to the complexity of the design of network services and the forecasting of future trends and scales of usage. However, new or emerging communication technologies have generally had their first brush with civilian use in the urban environment. In 1877, Alexander Graham Bell, along with his two financiers, Gardiner Hubbard and Thomas Sanders, formed the Bell Telephone Company to exploit his patented telephone technology. The first

telephone exchange, operating under license from Bell Telephone, opened in New Haven, Connecticut in 1878. Within three years of the inauguration of the telephone service, Bell telephone exchanges were in place in most major cities and towns across the United States. Cities and towns provided a critical mass of potential users within a limited physical space to allow for the diffusion and adoption of a new technology. The limited range of the initial telephone system was coincidentally suited to the urban space. The same is true today as the urban space, as opposed to suburban or rural environments, provides opportunities for many commercial and social applications as well as critical number of potential adopters of ad hoc networking technologies.

The effect of the diffusion of new technologies on cities and urban spaces has been of interest to researchers for many years [Gra96], [Mos99], [Tow00], and [Tow01]. Sociologists and urbanists have many divergent views on the impact of telecommunications systems on the fabric of the urban environment and the behaviour of the city dweller. [Mos99] explores the dramatic impact that information systems are having on the spatial organisation of activities in large metropolitan areas, increasing the complexity and type of interactions between individuals, firms, telecommunication systems and the built environment. The author argues that telecommunication systems are blurring the separation between home and work, changing the function of offices, cars, public parks and streets. Some of these changes may be seen in the move of banks from limited-hours operations on the main street to the provision of twenty-four hour services online. [Mos99] postulates that the creation of environments where humans can interact with inanimate objects, as may be facilitated through ad hoc networks, will turn such spaces into fully-fledged intelligent environments that will allow for the sharing of such pertinent information that may improve the quality of life for the users. Examples cited by the author included scenarios in which drivers stuck in traffic congestion may query and share information to less salubrious examples that illustrated how drug dealers and prostitutes may exploit such systems.

In [Tow00], it is argued that the widespread use of communications technologies is quickening the pace of urban life. The author contends that there has been an increase in the 'metabolism' of urban systems, i.e. the rate at which people interact with each other, make decisions and execute actions. This metabolism is tied to the decentralisation of cities as information networks are created in a distributed way to facilitate demands that are increasingly unconstrained by location and time. Decentralised systems create a myriad of new interactions and potential interactions between individuals. It is also suggested that urban planners have not taken account of the profound impact of new technologies on business and personal life. The placing of 2G and 3G antennas in urban locations to meet increasing demand is seen as a cosmetic response to the challenges posed. Traditionally, planners thought of the city in spatial terms: i.e. where are people and things located, how does this consequently shape their activity patterns? Modern telecommunication systems increasingly free people from spatial or temporal constraints. Where once one may have only accessed the Internet at the office PC or made a personal phone call at a payphone or on the home landline, it is now possible to access many services with increasing temporal and spatial freedom and at one's own whim.

A wide ranging exploration of the relationship between cities and communications technology is found in [Gra96]. The authors of 'Telecommunications and the City' highlight the social inequality associated with the Internet and most conventional hierarchical technologies, especially in advanced industrial cities. As urban society becomes more and more reliant upon telecommunications, the authors suggest that the city-

telecommunications interaction is the prime example of a society-technology relationship. They have sought to address the questions that arise when cities are increasingly composed of virtual companies, virtual communities and electronic territories. The result appears to be that there are both utopian and dystopian views of the effects of telecommunications on the city. As with many observers [Gra96] holds that technologies polarise society into the haves and the have-nots. Western cities have become increasingly de-industrialised as market forces have driven low-skilled labour-intensive plants to cheaper regions, away from the industrial heartlands of old. Thus, the voids left in the city fabric have increasingly become centres for retail, leisure and information-rich activities. Such revitalised spaces provide for the development and enabling of new applications and services that enrich the experience of shoppers, visitors, workers and the general urban dweller. Even though the impact of, and reliance upon, telecommunication systems has become more pervasive, [Gra96] maintains that urban planners are more concerned with shaping transport systems to meeting the public need as they are services that are visible to all. Telecommunications systems are a hidden public service and may be neglected to a greater extent than roads, schools and hospitals. While there is an outcry when roads are congested and trains are overcrowded, one rarely, if ever, hears of public outrage about an oversubscribed aDSL 'exchange' or patchy GSM signals in many urban blackspots. Urban policies are not developed to provide comparable qualities of service in both the visible and invisible services. Ad hoc networks, as extensions to existing infrastructures, allow communities of users to fill voids that either commerce or government will not service. Furthermore, the freedom of ad hoc networking technologies from the heavy regulation that exists for most centralised telecommunication services allows users to dictate the types of services that they want and the types of services they provide.

One other significant point that is raised by [Gra96] is that of the relationship between time, space, cities and technology. After Descartes, space was treated as an absolute object within which human life was played out. Space and time were viewed as mere environments in which social conduct is enacted. Many theories that were developed for the understanding of the growth of cities and urban systems placed a priority on the need to move people and goods as being the limiting factor defining the development of cities. The central tenet of the argument was that travelling across the Euclidean plane was expensive in both time and money terms. Services, from shops to stock markets, were spread out over space in proportion to population centres so that consumers could access them efficiently. Planners assumed that there was a decreasing interaction between people and services with increasing distance, and this approach shaped the growth of cities. As noted in [Gra96], the effects of new telecommunications systems have been interpreted within the context of the objective approach to space and the invariant notion of time. Telecommunications systems are still viewed as 'information highways' which are laid out across space and are treated analogously to transport networks. The authors argue that this view of telecommunications systems has 'straight-jacketed' new thinking on the role of such systems in relation to restructuring space. They further argue that if one views telecommunications systems as distance-shrinking technologies, one misses the point that such systems actually render distance meaningless rather than only reducing its degrading effects. Ad hoc networks allow people and objects to be networked at anytime and anyplace.

Community or non-Telco led telecommunication systems are already evident. These non-socially-polarising and rather more inclusive independent standalone systems attempt to address some of the problems and restrictions associated with centralised authoritative communications systems that have been subject to the

dead-hand planning of governments and corporations. Many college campuses have IEEE 802.11-based infrastructure wireless networks and more hotels, airports, cafes and shopping malls are offering such services on a per-fee basis. Emerging 2.5 and third generation telephones are enabled with packet-based communication mechanisms (GPRS, CDMA2000 etc) and many are equipped with Bluetooth. Gradually, a vista of more pervasive consumer data networking is opening up. Campus LANs and Bluetooth-based PANs offer the possibility of networking in the absence of a Telco and therefore a third-party billing authority is not a feature. The evolution of these standards may well see the realisation of open standard ad hoc networking protocols being adopted across a variety of platforms and transmission media, through a variety of business and community-led initiatives.

At a more technical level, the urban environment is a challenging space for any wireless technology. The challenges vary over space and time and are compounded by the variability and unpredictability of users' movements and data demands. A cursory look at any urban landscape reveals a mosaic of physical spaces. Open city parks are bounded by busy roads, lined with trees and terraces. Pedestrianised streets are flanked by shops, cafés and car parks. Each physical space brings with it a different obstacle for an ad hoc networking device. Walls, trees and vehicles interfere with radio signals, causing multi-path fading and other hurdles that must be overcome by more robust modulation schemes such as OFDM. Planned, infrastructure-based systems such as GPS and GSM, which may be used to augment the ad hoc routing services, are sometimes available intermittently and with varying levels of quality.



Figure 2-7 A Pedestrianised Shopping Street

The review of ad hoc routing protocol performance in this chapter has highlighted the varying success and efficiencies of ad hoc routing protocols in a variety of settings. Certain routing protocols are shown to be suited to particular combinations of traffic loads and node mobility characteristics, among other parameters. While the urban space is a diverse and heterogeneous patchwork of spaces distinguished by diverse characteristics, the global urban space may be broken down in smaller local homogeneous regions. Such local regions have specific demands and networking characteristics as a result of the activities that are played

out in their distinct physical space. A single generic ad hoc routing protocol would not be optimally suited to such a diverse and dynamic environment.

An example of a typical urban space is described to elucidate the elements of user diversity and networking diversity outlined in the previous paragraphs. A pedestrianised street adjoining an urban park is an example of two contiguous but uniquely distinguishable spaces. While there are few physical barriers between the two areas, the behaviour and experience of people in the two spaces is quite distinct. These two areas are worthy of further discussion and exploration as they provide a context in which to examine further work in this thesis.

A *pedestrianised shopping street*, such as Dublin's Grafton Street, is possibly one of the densest environments in an urban space in terms of humans and associated retail and leisure services. Shoppers cram into such streets, moving about in bursts and flows as they enter and leave shops or congregate around buskers and shop windows. There is a mix of purpose driving the people that circulate in such a space. Some people meander in and around stores or stop to drink a coffee. In such a setting the types of possible applications may vary greatly. Shoppers might query stores on the brands that they stock and their inventory without setting a foot inside the door. Department stores may target customers who flag themselves as loyal customers with advance information on sales and special offerings. The types of traffic in such an environment may be sporadic and short-lived. Shops, shoppers, cafés and customers are only briefly associated for the purposes of sharing and querying information as they come into proximity with one another.



Figure 2-8 An Urban Park

In a *recreational city park*, such as Dublin's St. Stephen's Green, one may expect a very different environment to that found in a busy retail space such as the street described above. Urban parks and squares are spaces that have been designed as relaxing spaces. They don't team with milling people in the same frenetic way that a mall or a street does. People have space to relax in a park as the pace of life slows down a notch or two. This is reflected in the way that such a space may be characterised in terms of the 'density' and mobility of humans. The density of humans in a park will generally be moderately sparse while the mobility of visitors is quite slow, if they are not entirely stationary. Such a relaxed setting as a park allows people to engage in more

thoughtful and long-lived network transactions. People may surf the Internet or check their e-mail account. There is time to browse news sites or municipal information portals. The scenario in which the user is situated influences the type applications and services they will use and the time they will spend using them. A pedestrianised street is not the place to idly book a holiday online, whereas the park bench is not the place from which to browse store offers.

All in all, the urban space should provide a fertile environment for the growth of ad hoc networking technologies in years to come. It is one of the prime settings in which civilian, commercial and open mobile wireless ad hoc networks may thrive as there will always be a critical mass of potential users, services and applications to stimulate the formation of a network. However, the diversity of such a space would challenge existing approaches to routing in ad hoc networks as routing protocols have not been designed with such diverse and dynamically varying scenarios in mind.

2.7 Summary

This chapter has demonstrated that there is much diversity in the field of ad hoc networking research, ranging from the choice of available radios and MAC schemes on the one hand to the variety of applications and services that are proposed on the other. That diversity and variety are also present at the core of the ad hoc network, specifically at the network-layer. Many ad hoc routing protocols have been proposed and developed over the last number of years. It is evident that a few of these multi-hop ad hoc routing protocols have proven themselves as they have matured and moved through the development, evaluation and standardisation tracks of the global ad hoc network research community.

These protocols have emanated from many different sources; some taking their design influences from the routing protocols used in conventional wired networks, while others present completely novel techniques that have been motivated by the intrinsic characteristics of the mobile ad hoc networking domain. The evaluation of these protocols has clearly demonstrated the disparity of the protocols' performance profiles when challenged under a range of differing networking conditions.

Whatever the provenance or performance of the protocols, it is clear that no single ad hoc routing protocol will become the preferred and unequivocal choice of the research community; instead there will most likely be a number of competing standards that work well under different networking conditions.

While the diversity of choice of ad hoc networking solutions is quite clear, the future context of ad hoc networking is more uncertain and conjectural. The place for mobile ad hoc networks is assured within the confines of the military context; much of the flourishing contemporary research in this area is stimulated by the needs of the military. As with most technologies that originate from those quarters, ad hoc networks are sure to be adopted in commercial and civilian contexts. In an open, civilian environment, mobile wireless ad hoc networks may be seen as an emancipating technology; the constraints of hierarchy, of centralised authorities and forward planning are jettisoned in place of autonomous, robust, self-organising and distributed traits that allow nodes to form demand-driven networks that are independent of space and time. In open environments, devoid of rigid social hierarchies, autonomous ad hoc networks will be required to

manage themselves in the absence of any external controls. Such environments are also likely to exhibit different networking conditions from time to time and from place to place.

It is apparent that within such a demanding and heterogeneous mobile ad hoc networking context there is an argument for flexibility at the network-layer. In a general sense, flexibility is an umbrella concept that encompasses such design features as systems, structures, reconfigurability and adaptability. The need for flexibility at the ad hoc network-layer is motivated by a number of factors. Flexibility is motivated by the diversity of choice of ad hoc routing protocols, the disparity of their performance profiles and the challenging nature of an open, autonomous ad hoc networking environment.

Flexibility is required to address the coexistence of multiple standardised multi-hop ad hoc routing protocols. In a networking environment that is not monopolised by any one protocol, the networking system should be flexible enough to accommodate and leverage the availability and use of multiple standardised protocols.

Flexibility is required to address significant changes in the networking conditions experienced by autonomous nodes. The variation in the performance profiles of some mature protocols strongly indicates that no single protocol will deliver an optimum performance under *all* networking conditions. A flexible networking system should be able to engage the protocol most suited to the prevailing network conditions.

Finally, flexibility is required to address the challenging nature of an open, autonomous ad hoc networking environment. Such an environment is characterised by the occurrence of events such as nodes migrating from one network to another, network partitions, network mergers and initialising networks. As ad hoc routing protocols have been designed with such events in mind, a system that facilitates the use of multiple protocols in diverse environments should itself not be constrained by such events. The next chapter proposes a definition of a flexible ad hoc networking system and identifies the design challenges that arise in creating such a system.

3 Requirements of a Flexible Multi-Protocol Network-Layer

3.1 Introduction

The notion of a flexible ad hoc networking system was suggested and extensively motivated in the last chapter. On that basis, this thesis proposes a move away from the traditional approach to network-layer design, as found in both wired static networks and existing wireless ad hoc networks.

The routing protocol is at the core of the network-layer. Conventionally, the routing protocol is considered to be an invariant with regard to the layer structure. The standard approach to the design of ad hoc network routers and networks has been to choose a single protocol, e.g. AODV, DSR, TORA or a location-based protocol such as LAR or DREAM, prior to the deployment of the node and the network. As with other layers of the communications stack, with the exception of the application and session layers, the choice of what particular service to use is made before the node is deployed and would rarely, if ever, be replaced or upgraded.

It is therefore proposed in this thesis that a node be designed to have access to a suite of ad hoc routing protocols that it may employ, as appropriate, in the course of its existence. This chapter defines the core of a multi-protocol ad hoc networking system and lays out what would be expected of the network-layer in such a system. As will be seen, there is a step-change in network-layer design, from a monolithic and preconfigured

structure to a flexible multimodal structure which introduces a large number of demands on the network-layer.

3.2 Flexible Multi-Protocol Network-Layers

In terms of the research carried out in this thesis, a flexible multi-protocol network-layer is defined as a network-layer that allows the nodes of an ad hoc network real-time access to a suite of routing protocols. The purpose of such a network-layer is to facilitate the real-time configuration and reconfiguration of nodes in response to naturally occurring network events and changes in the networking conditions; such a level of flexibility was previously not possible. In order to highlight the points where flexibility is required in such a multi-protocol network-layer it is useful to depict a typical day in the life of autonomous nodes in an average urban environment. The example used revolves around an urban *Park* and an adjoining pedestrianised shopping *Street*, as illustrated in Figure 3.1 below.

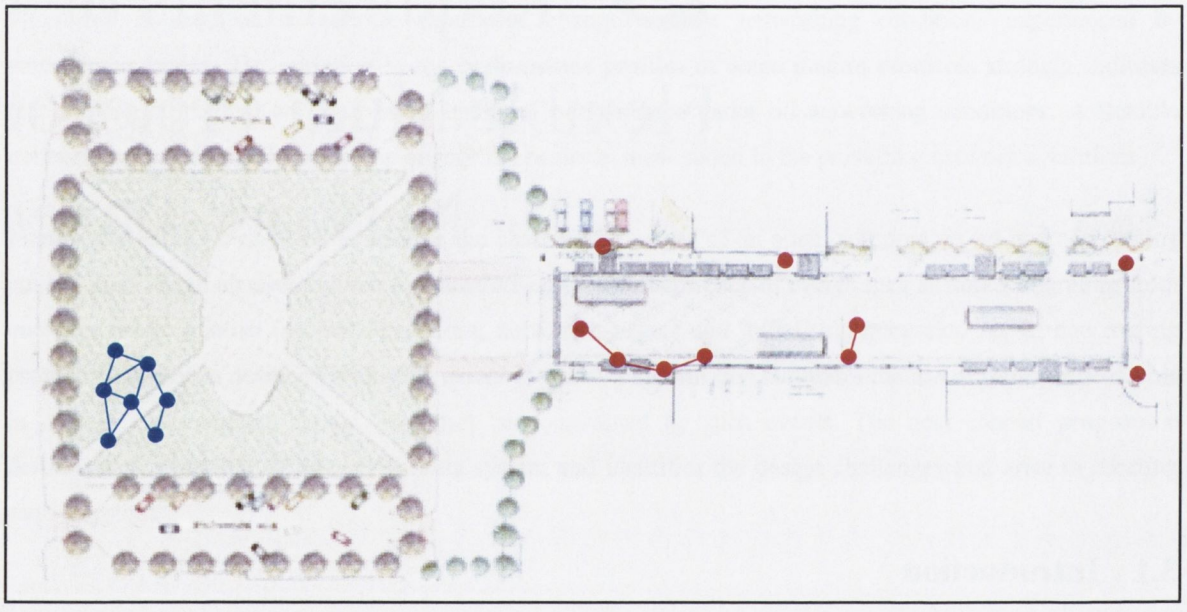


Figure 3-1 Illustration of a Park and an Adjacent Street: Morning

Morning: A group of students agree to meet in the Park for a game of football and a little study before their afternoon lectures. When they arrive in the park they turn on their mobile devices: laptops, tablet PCs and PDAs, to share notes and collaborate on assignments from the day before. (It is acknowledged that the foregoing list of contemporary computing devices is sure to be outdated by more ambient devices and appliances in the near, but unforeseeable, future.) Each of the devices is equipped with the latest flexible ad hoc routing software, allowing each node to choose from a selection of protocols and services. The ability of nodes to seek software updates from time to time, thereby changing or upgrading their suite of protocols, is not a consideration of this thesis. The group of nodes ***initialise*** themselves to use a common ad hoc routing protocol and form an ad hoc network, the blue network in Figure 3.1. Meanwhile on the Street, some early morning shoppers have arrived to look for bargains and do some window-shopping. An ad hoc network already exists here as the semi-permanent commercial nodes in the shop windows and the municipal authority nodes in the lampposts have already initialised. The shoppers' nodes ***join*** the network on the street, the red network in Figure 3.1

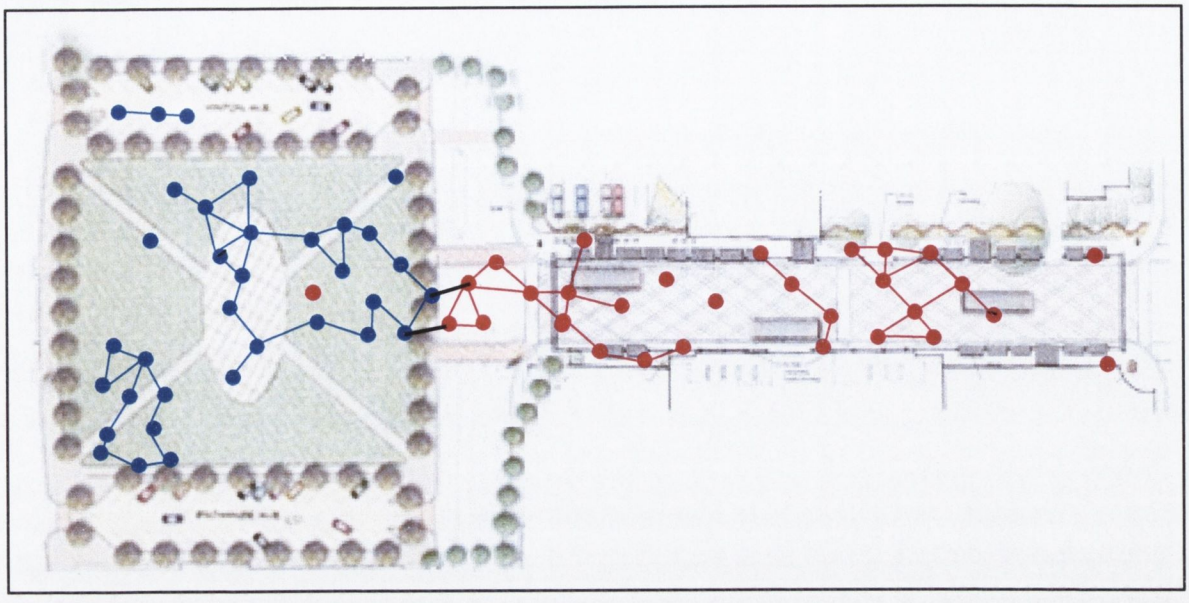


Figure 3-2 Illustration of a Park and an Adjacent Street: Afternoon

Afternoon: As the day progresses, more people come to the Park for lunch. They may come from the Street where they have been using the red network. As they come into the Park their nodes must ***migrate*** to the blue network. Whereas initially there were only a few students in the Park, the numbers in the Park have now increased dramatically and the protocol that the blue network is using is starting to fail. The blue network, after observing and analysing the change in the network dynamics, ***reconfigures*** itself to use another protocol. While in the Park, some people want to check out the latest offerings from the music stores and bookstores on the Street. As they are part of the blue network in the Park and the music store is using the red network in the Street, the red and blue networks must ***internetwork*** to allow communication between the Park and the Street, as illustrated by the black lines connecting the red and blue networks in Figure 3.2.

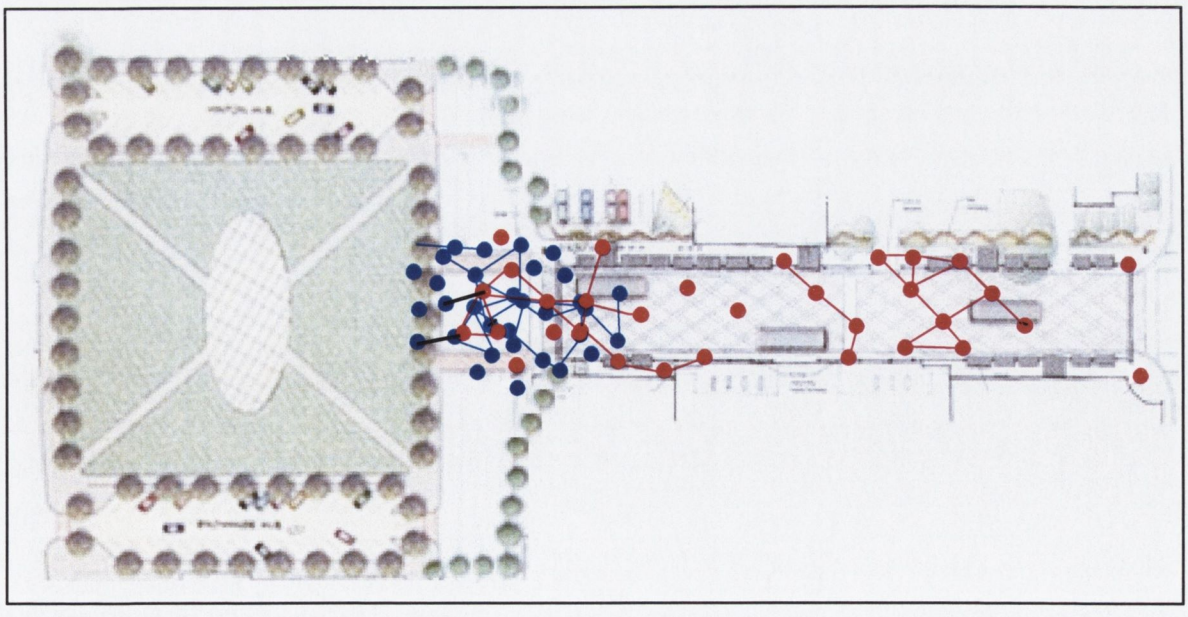


Figure 3-3 Illustration of a Park and an Adjacent Street: Evening

Evening: As the Park closes at six o'clock people start to head for the Street. As the blue network moves in to the Street the two networks have to ***merge***, as illustrated in Figure 3.3. At each of these points in the life of a

node and a network, the system is required to enable both the networks and the nodes to adapt to the prevailing networking conditions and composition in a timely manner without any manual or external interference.

The goal of this thesis is the design of a multi-protocol network-layer for an ad hoc network that facilitates the type of behaviour as depicted in this example. The scenario outlined here clearly identifies the stages in the life-cycle of an ad hoc node at which flexibility in the network-layer is desirable (i.e. initialisation, node join, network mergers, node migration, internetworking and adapting to changed conditions). The following subsections take each of these stages in turn and give more detailed descriptions of the type of flexibility that is proposed.

3.2.1 Flexibility to Enable Auto-Configuration: Node Initialisation and Node Join

It is proposed that a node should be able to auto-configure itself at the network-layer. The term auto-configuration, as used in the context of this work, describes a node's ability to choose a routing protocol at runtime so that it may form a network with other nodes. The configuration of other network-layer parameters, such as a node's address, is not at issue in this work.

It should be remembered that multi-hop peer-to-peer communication is only possible when remote nodes can exchange routing information by using a common routing protocol. As conventional routers are statically configured with a single invariant ad hoc routing protocol, network-layer auto-configuration, with regard to the routing protocol, has not been an issue heretofore. In conventional routers, each node simply follows the boot-up procedures mandated for the protocol statically deployed in the network-layer. In a network of flexible multi-protocol nodes the adoption of a common routing protocol at boot-up time is not so trivial. The specification of a default option is difficult, if not impossible, as nodes may be deployed at different times and by different vendors and user groups.

In closed networking systems, such as may exist in the military or commercial domains, access to the network can be tightly controlled. An army battalion may number up to a thousand soldiers depending on the number of constituent companies, translating into a predictable number of ad hoc nodes. In such scenarios, nodes are admitted to the network in an organised way. There may also be a pre-existing association between the nodes in such a network, mirroring the social or commercial associations of their owners/users. Such an association may take the form of a network or partition identification number.

However, in an open ad hoc network there is no requirement for a pre-existing association between the participating autonomous nodes. As the nodes are autonomous, they can form networks whenever they need to. From time to time and from place to place a node may cooperate with various other nodes to form distinct standalone networks that serve diverse and transient purposes. In such an environment, where multiple standalone networks may be formed in isolation from one another, it is important that such standalone networks are identifiable, both from an internal and external viewpoint. Nodes participating in the network should be enabled to recognise the nodes they are collaborating with and should also be enabled to recognise foreign nodes and networks, i.e. nodes and networks with which they have not yet formed an association.

Auto-configuration should readily facilitate the existence of multiple independent networks so that each network may cope with future network events such as migration, mergers, partitions and reconfiguration.

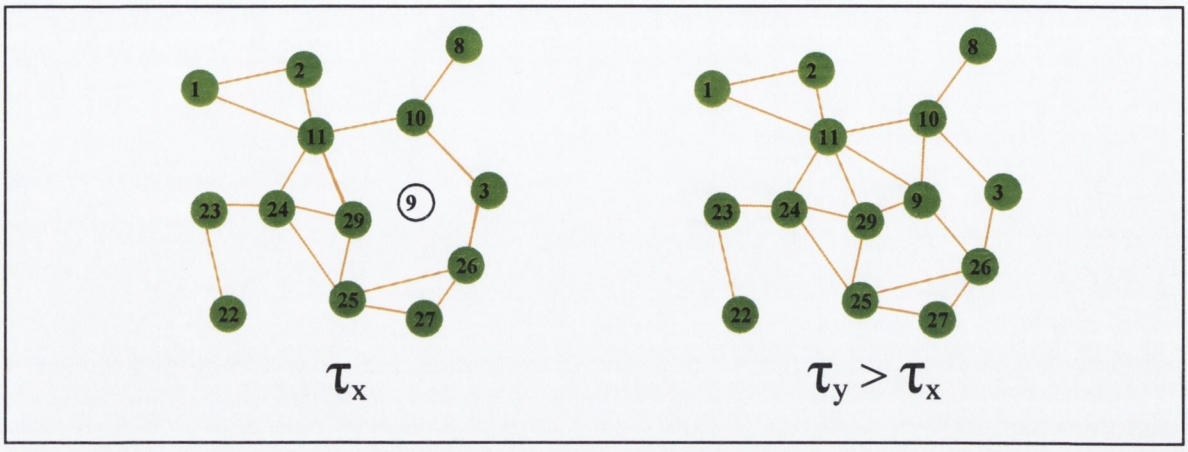


Figure 3-4 A single node boots up and auto-configures itself

It is envisaged that nodes will generally auto-configure in two ways. Firstly, a single node may boot up either in the presence or absence of a network or other node. If a node boots up in the presence of an existing network, it may join that network by configuring itself accordingly and associating itself with that network. Figure 3.4 depicts node 9 at time τ_x when it is being booted. The node has not yet configured itself and cannot join the network. By time τ_y node 9 has configured itself appropriately and has joined the green network.

The second common scenario is group initialisation. This occurs when a group of nodes simultaneously boot to form a network in order to collaborate and communicate. In this case, the challenge that the network faces is in negotiating and choosing a protocol in the absence of functioning peer-to-peer communication services. Figure 3.5 depicts a group of seven nodes that boot up within a short time before time τ_x. As none of the nodes are configured they cannot perform multi-hop peer-to-peer networking. By time τ_y they should have collaborated to configure themselves so that they can form a network.

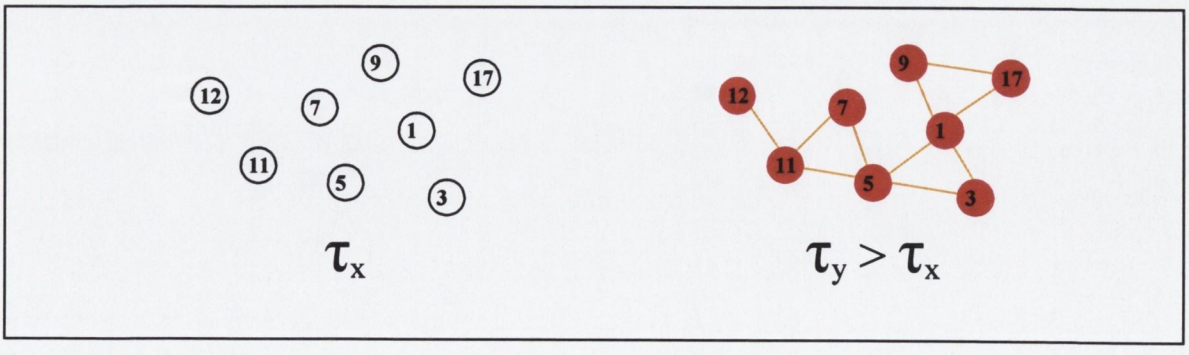


Figure 3-5 A group of nodes boot up and auto-configure themselves

It is envisaged that the process of initially auto-configuring a node should be *proactive* in nature, i.e. a node will actively seek to be configured when it boots up, regardless of whether or not the node is to be used immediately. It is reasonable to assume that when a node has been turned on and booted, that it will want to

join a network immediately as an un-configured node cannot participate in a network. This is also the case in the group situation. When nodes boot-up in a group situation, as depicted in Figure 3.5, they may not be within direct communication range of each other. It is possible that some nodes may be *remote* from other nodes in the group. Until the nodes' network-layers are configured they may not communicate in a multi-hop peer-to-peer manner. The lack of such communication facilities inhibits the ability of the nodes to organise themselves.

As is obvious from the examples, much of the proposed flexibility in this ad hoc networking system revolves around the ability of the nodes to access and use a number of protocols, in contrast to conventional single mode routers. The network may be globally configured in one of a number of *multiple possible states*, which are *equivalent* and *interchangeable*. As such, the network-layer of each node may be configured to use a range of different protocols, e.g. protocol A or protocol B or protocol C. Each protocol, whether it is protocol A, B or C, is equivalent in terms of the fact that it provides *a* multi-hop routing service, but different in the manner in which it delivers *that* multi-hop routing service. No protocol has any natural precedence over any other protocol at boot up time; there is no preferred initial state of configuration. By initial precedence or preference, it is meant that in the absence of experience of the networking dynamics it is not possible to know, in advance, which protocol will be best suited to the network when it is active. The primary goal of network-layer auto-configuration is to ensure that the participating nodes realise a *consistent global state* across the network. It is not the goal of the auto-configuration mechanism to focus on which protocol would be suitable to use under the existing networking conditions, as there would be no experience of such conditions at this point in the life-cycle of the node.

It is worth noting that the size of an open network is unknown, i.e. the number of participating nodes may vary from time to time in an unplanned or unforeseen manner. In keeping with the spirit of a truly *spontaneous* ad hoc network, it is not possible to ascertain the number of participating nodes in the network at any time. As the ad hoc network is not a closed system, nodes are free to join and leave at will. The joining and leaving of nodes has the effect of changing the size of the network in a dynamic and unpredictable way. The auto-configuration process should not be impeded by such uncertainty over the network size.

In essence, the objective of network auto-configuration is to ensure that a consistent global network state, i.e. a common routing protocol, is brought about by the participating nodes within a short time of them being booted up.

In summary, a flexible system that facilitates the auto-configuration of a node in any networking situation should be able to:

- Observe the environment the node is booting up in.
- Collaborate with other nodes, if any, so that a consistent state is brought about at the network-layer with regard to the ad hoc routing protocol configuration.
- Implement a decision, based on prior observation and possible collaboration, so that the node is configured and ready for use in a timely manner.

3.2.2 Flexibility to Enable Auto-Configuration: Node Migration and Standalone Network Mergers

After a node has initially auto-configured itself, it may begin to participate in the network. At this stage, the node may experience migration and network merger events. It is proposed that a node should be able to migrate between standalone networks in a seamless manner. The term node migration, as used in the context of this work, describes a node's ability to adapt the configuration of its network-layer, i.e. its routing protocol, route cache and other data structures, when it moves from one standalone network to another.

The ability to migrate is a necessary feature of a flexible ad hoc networking system as standalone networks may be formed in separate physical spaces using different network-layer configurations. It is envisaged that a flexible ad hoc networking system should allow standalone networks to be formed as and when they are required. It is also envisaged that autonomous standalone networks should be enabled to reconfigure themselves, as will be discussed in the final subsection, in response to observed networking conditions and are free to move in an unrestricted manner. As nodes move about a physical space in an autonomous manner, they may move from one network to another and the nodes should be enabled to adapt to observed changes in the state of the network enveloping them. Such adaptation should involve the nodes configuring themselves to the state of the network by adopting the routing protocol and data structures that are consistent with the new network's state.

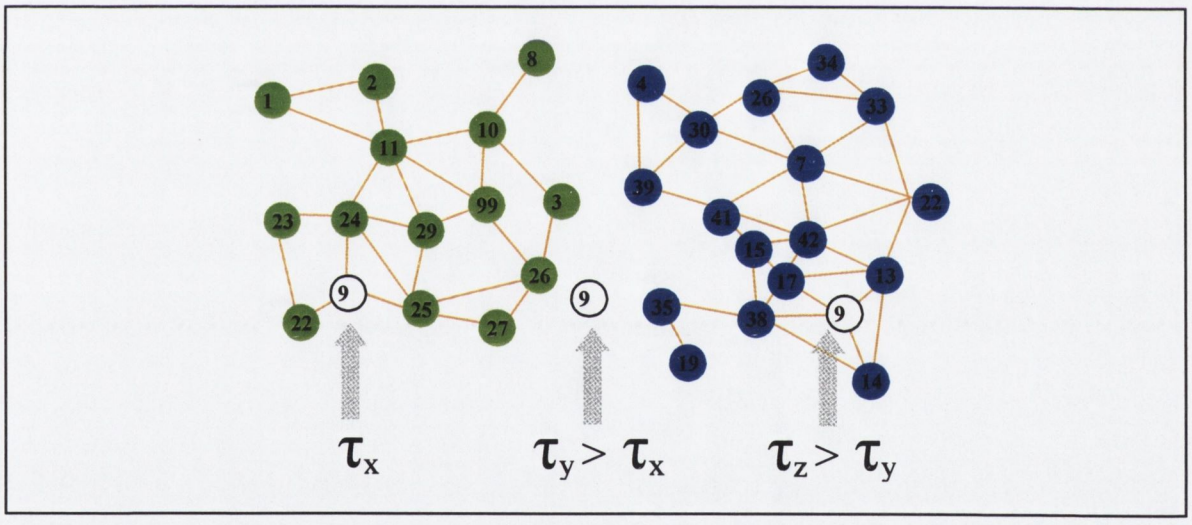


Figure 3-6 Node Migration between Two Standalone Networks

The importance of dealing with node migration stems from the fact that when a node moves from one area to another there may be a resulting inconsistency in the network state as the migrating node is out of step with the enveloping networking system. A migrating node is a node that becomes physically isolated from the network that it has formed an association with. The migrating node would have been '*in synch*' with the network it left but there is no guarantee that the network it migrates to will be configured likewise. The two parties to the migration event, the migrating node and the receiving network, should resolve the differences between their configurations and identities in a timely manner so that the state of a receiving network is not perturbed by the signalling of a misconfigured node and so that the migrating node may network.

An example of a scenario that demands such system flexibility is depicted in Figure 3.6, where two networks exist in close proximity. The green network is using the Protocol A and the blue network is using the Protocol B. A time τ_x, node 9 is situated within the physical area occupied by the green network and participates in that network by using the Protocol A. Node 9 starts to move and by time τ_y it has just left the coverage of the green network. By time τ_z, node 9 has moved into the area covered by the blue network. At this time, node 9 has physically migrated from the green network to the blue network and will be unable to communicate with the other nodes in the blue network as it is using a different protocol. A flexible system should enable node 9 to observe the changes in the state of the surrounding nodes and adapt its own state accordingly so that it may continue to network.

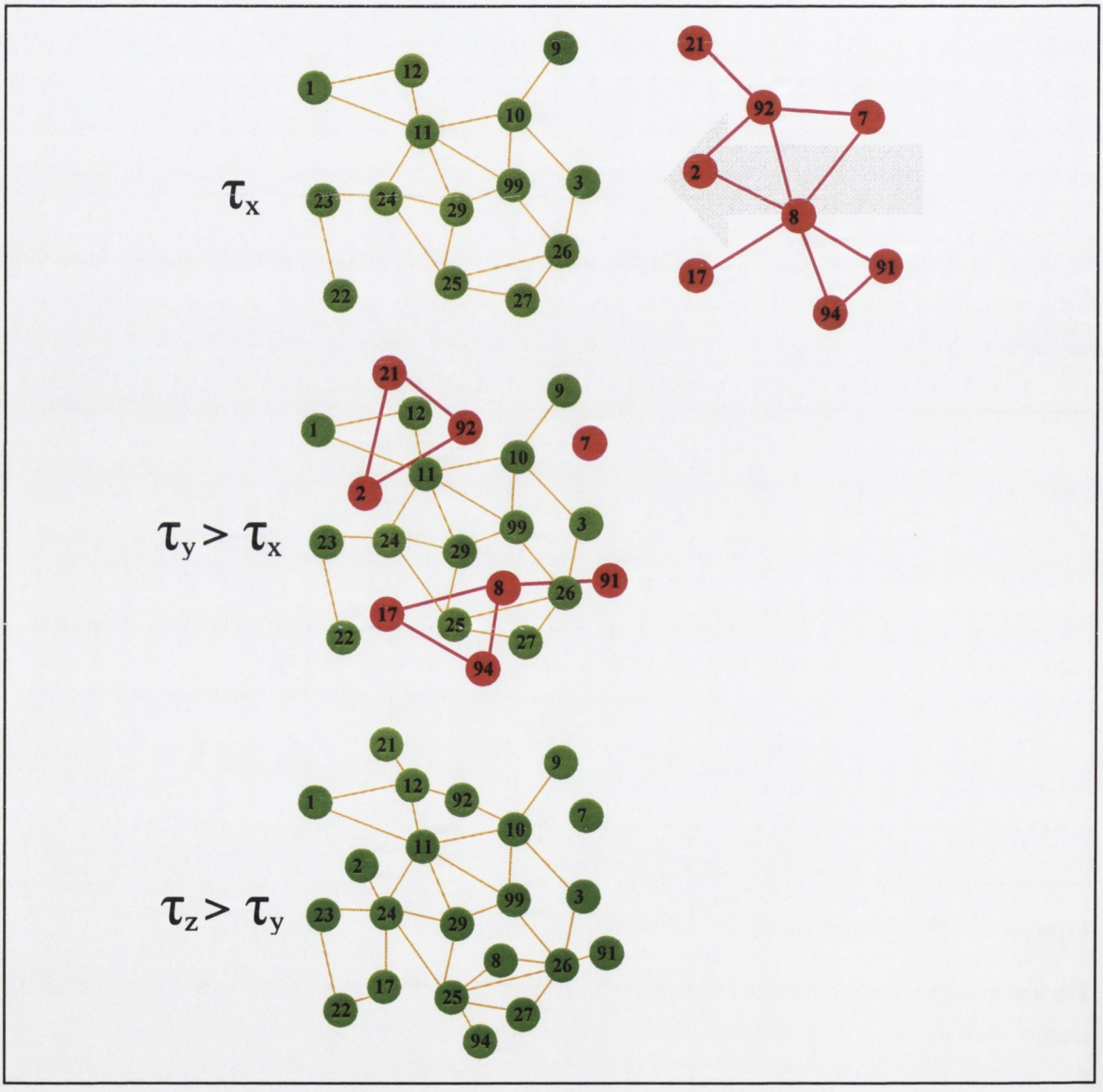


Figure 3-7 Complete Merger of Two Standalone Networks

It is also proposed that standalone networks should be able to merge. The notion of a network merger, as the term is used in this work, encompasses the ability of the networks, as manifested in their constituent nodes, to adapt the merged nodes of the two networks so that a consistent state exists in the merged nodes, allowing for peer-to-peer communication.

In some networking contexts a merger happens when the edges of two networks come into direct transmission range with each other. In this work, a network merger is considered to occur when two networks share the same physical space. The merging of multiple networks demands a different approach to that of node migration. To clarify, Figure 3.7 depicts two networks that go through the process of a complete merger. At time τ_x, both a green network and a red network exist, both operating different protocols. The red network is moving in the direction of the space occupied by the green network. A time later at time τ_y, the red and green networks occupy the same space and ether but they cannot communicate with one another as they are using different protocols. By time τ_z, the nodes of the red network should have adapted their configuration to use the green protocol, allowing the two former standalone networks to function as one homogeneous standalone network.

Instead of having multiple overlapping networks occupying the same ether, but using conflicting ad hoc routing protocols so that they cannot communicate in a multi-hop peer-to-peer manner, it is preferable that the merged nodes adopt a single protocol. In contrast to the situation that occurs when a node migrates, the nodes of merging networks are still part of their original networks unlike a migrating node that becomes isolated from its original network. The merged nodes should be enabled to take mutually acceptable actions that result in the merging of the two networks in such a manner that a consistent network-layer configuration persists in the merged section of nodes.

In summary, a flexible system that enables a node to migrate from one network to another should enable the node to:

- Observe that it has migrated from one network to another network.
- Evaluate the choices available to it that may enable it to participate in the other network.
- Implement a decision in a timely manner with minimal service loss, allowing the node to rejoin a network.

A flexible system that facilitates the merging of two standalone networks should be able the nodes to:

- Observe that a network merger is taking place.
- Evaluate the options available to both networks.
- Implement a mutually acceptable decision that results in the merged section of the network regaining a consistent state in a timely manner.

3.2.3 Flexibility to Enable the Ad Hoc Internetworking of Heterogeneous Standalone Ad Hoc Networks

It is proposed that two heterogeneous standalone networks should be able to internetwork. The term internetworking, as used in the context of this work, describes the ability of heterogeneous networks to route data between nodes in both networks.

Internetworking is a necessary feature of a flexible ad hoc networking system as two heterogeneously configured standalone networks that are within transmission range of each other should not be isolated from

one another, in terms of multi-hop communication, by virtue of their network-layer heterogeneity. In order to allow the two networks to internetwork, the communication of routing information and data between the two networks should be facilitated. Gateways, bridges, interior routing protocols and exterior routing protocols are features of conventional internetworks. However, their applicability to the mobile ad hoc networking domain is limited. The autonomous and mobile nature of ad hoc networks demands a more dynamic and distributed solution to the problem of ad hoc internetworking.

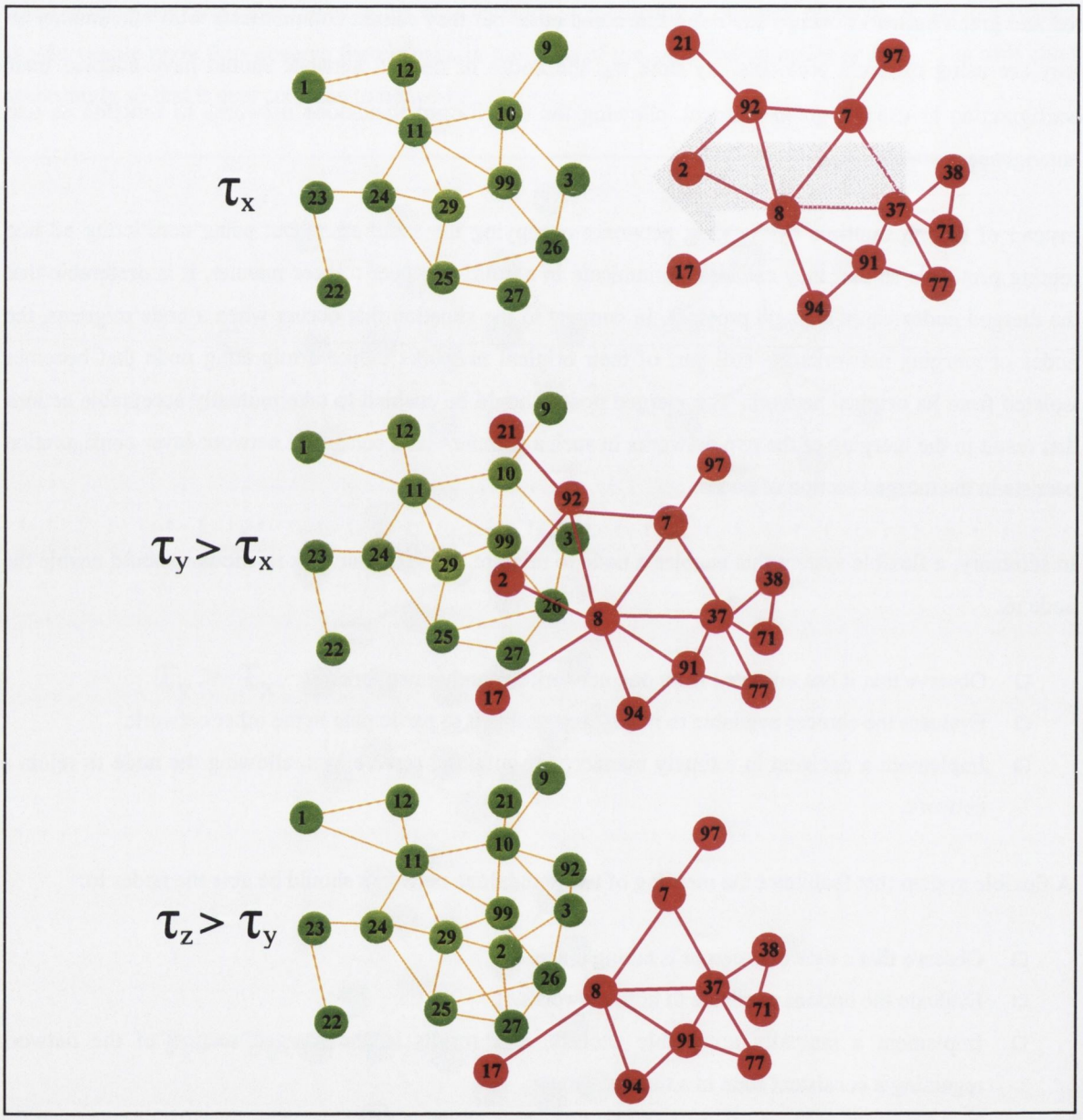


Figure 3-8 Partial Merger of Two Standalone Networks

A flexible system that enables internetworking should dynamically enable nodes to act as *bridges* between two contiguous regions. Such bridge nodes, in addition to having normal routing functionality, should be capable of facilitating internetworking between the two heterogeneous networks. Once again, such flexibility is dependent on each potential bridge node being able to determine the need for bridging capabilities between two contiguous, yet heterogeneous, regions of nodes.

An example of a scenario that necessitates internetworking capabilities would possibly arise following on from the merging scenario outlined in the previous section. Figure 3.8 depicts a networking scenario that is similar to that already depicted in Figure 3.7 except that in this example the two standalone networks do not merge completely. The red network partly moves into space and ether occupied by the green network at time τ_y. Unlike the example depicted in Figure 3.7, not all of the nodes in the red network share space or ether with the green network. By time τ_z, the portion of the red network that overlaps the green network has adapted to the observed network composition change by adopting the configuration of the green network. Such action would result in the creation of a single internetwork of two heterogeneous regions of nodes where originally there were two unconnected standalone networks.

For whatever reasons a heterogeneous network arises, internetworking is needed to enable communication across the entire network. The green network in Figure 3.9 may be using the Protocol A while the blue network is using the Protocol B. A flexible network-layer should allow a node within direct communication range of both networks, node 50 in Figure 3.9, to provide internetworking facilities.

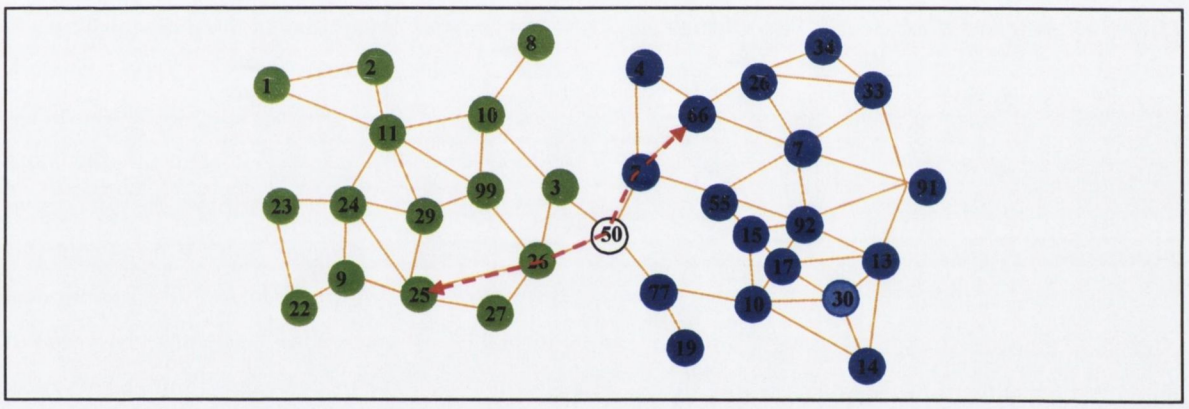


Figure 3-9 Internetworking of Two Heterogeneous Standalone Networks

In summary, a flexible system that facilitates internetworking between two heterogeneous networks should allow a node to:

- Observe the presence of multiple contiguous heterogeneous networks.
- Evaluate the necessity to provide internetworking facilities.
- Implement the necessary modifications to the node's network-layer structures to allow it to facilitate internetworking.

3.2.4 Flexibility to Enable the Reconfiguration of Networks

The final proposal of this thesis is that nodes should be able to reconfigure the network-layer in response to changes in the networking conditions, e.g. node mobility, node density and traffic. Specifically, nodes should be able to reconfigure dependent features of the network-layer, namely the routing protocol in use.

The need for nodes to be enabled to reconfigure themselves at the network-layer is motivated by the fact that it has been demonstrated in the literature presented in the last chapter that various protocols are suited to disparate networking conditions. Whilst acknowledging the equivalence of the routing protocols, in that they

all provide a multi-hop peer-to-peer routing service, it is apparent from the published literature that the efficiency with which different protocols deliver that service is dependent on the underlying networking conditions. It is proposed in this thesis that a flexible network-layer should enable networks to utilise the optimum routing protocol for the prevailing networking conditions.

The fundamental requirements of such network reconfiguration are threefold. Firstly, nodes should be enabled to observe and analyse the changing networking conditions in which they are operating as the efficiency of routing protocols is affected by a number of different factors such as node degree and node mobility. Secondly, nodes should be able to collaborate with each other to ascertain whether or not there is a requirement to reconfigure the network. Such collaboration may involve the exchange of information about the networking conditions or the decision-making process. As previously stated the routing protocol used by a node is a dependent variable of the network-layer and cannot be changed at a node in a unilateral manner. Finally, nodes should be able to act on a decision in a timely and organised manner, in conjunction with other participating nodes, to reconfigure the network to use a more suitable routing protocol such that the new node configuration is more suited to the prevailing network conditions.

Given the open nature of the proposed ad hoc network, the more specific objective of this component of the flexible multi-protocol layer is to allow *groups* of nodes, which are already collaborating to form a network using some ad hoc routing protocol, to choose the routing protocol most suitable for the region of the network that they occupy at that time. As networking conditions are unlikely to be uniform across the entire expanse of a realistic ad hoc network at all times, the flexible network-layer should allow the network to function as a patchwork of heterogeneously configured regions. The term *region* is used throughout this thesis in place of the term *partition* as the term partition connotes a stronger and more distinct physical separation between various network elements. In this thesis, while the network may be divided by virtue of heterogeneity at the network-layer, the division may be overcome by technical innovation, unlike a physical network partition.

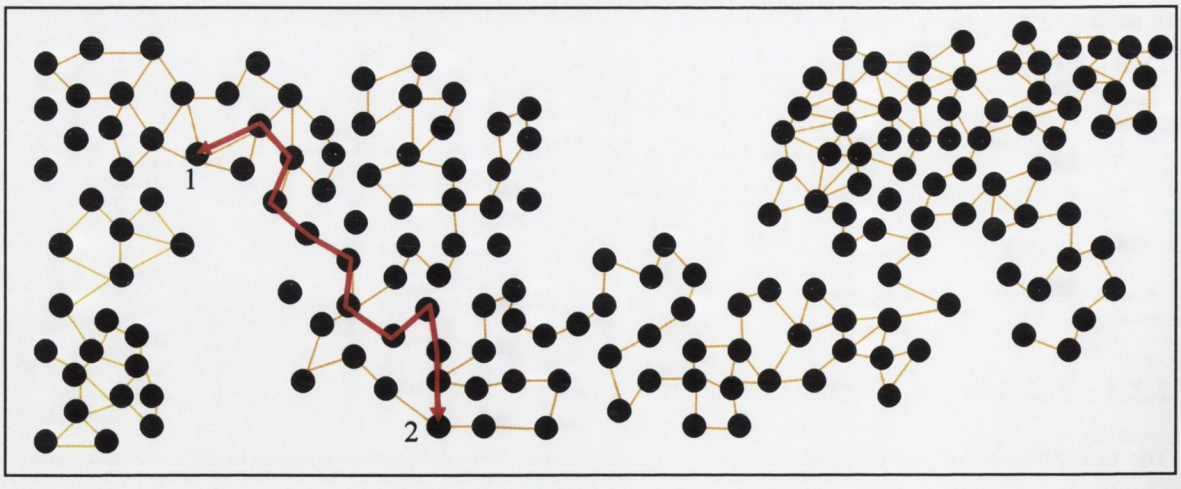


Figure 3-10 A Network of Statically Configured Single-Protocol Nodes

To illustrate the general concept of a flexible heterogeneously configured network, the following diagrams contrast the differences between existing networks and the proposed network. In Figure 3.10, every node uses the same routing protocol, as is typical of a standard ad hoc network. This enables Node 1 to communicate

with Node 2, from end-to-end, by accessing the multi-hop routing services provided by the intermediary nodes, as illustrated by the thick red line in the diagram. The thinner orange lines denote connectivity between nodes.

In a flexible network that enables nodes to take account of regional differences in networking conditions, the nodes participating in an end-to-end communication session may not be configured to use the same routing protocol, as illustrated in Figure 3.11 where Nodes 1 and 2 are in adjacent, heterogeneously configured regions. The flexible multi-protocol network-layer, by preserving the underlying and unifying unreliable packet-switching service, enables the seamless internetworking of heterogeneously configured regions of nodes that reflect the underlying networking conditions. The reconfiguration component, in conjunction with the other features of the flexible multi-protocol network-layer, enables the network to suitably configure itself on a regional basis. The flexible attributes of this network-layer enable the network to bend, i.e. accommodate different conditions by using different protocols, without breaking, i.e. fail to enable multi-hop peer-to-peer communication because the routing protocol cannot cope with the prevailing conditions.

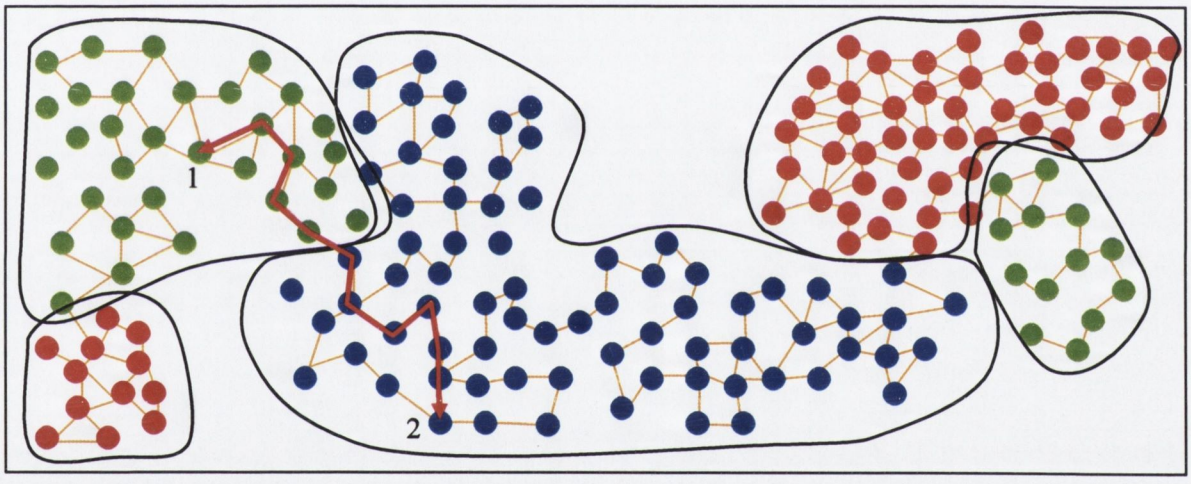


Figure 3-11 A Network Composed of Heterogeneously Configured Regions

In summary, a flexible system that facilitates the reconfiguration of a dependent feature of a node's network-layer should enable the node to:

- Observe and analyse the prevailing networking conditions.
- Collaborate with other nodes to determine if another routing protocol is more appropriate.
- Implement a decision, based on prior observation and collaboration, so that the node/network is reconfigured in a timely manner.

3.3 Summary

The notion of a flexible ad hoc networking system, introduced in this chapter, is predicated on the concept of a multimodal ad hoc network-layer. Such a network-layer would, unlike conventional routers, be designed around a suite of standardised ad hoc routing protocols that have been shown to have different performance profiles under different networking conditions. As discussed, the flexible ad hoc network-layer would enable nodes to *auto-configure* (initialise, join, migrate, merge), to *internetwork* between heterogeneous regions of

the network, and to *reconfigure* regions of nodes in response to changes in the observed networking conditions. To enable this level of functionality a node in the network must have the ability to observe its operating state and its environment as the network, through its nodes, only adapts to changing conditions if it is able to observe those conditions. A node should then be able to make a decision, based on its observations (and where relevant, in collaboration with others), on which of the available ad hoc routing protocols to use. It is clear from the discussion in this chapter that a truly flexible multi-protocol network-layer must contain a high level of functionality and sophistication. In designing such a layer, there are many challenges relating to the specific characteristics of the ad hoc network that must be taken into account. The remaining chapters in this thesis deal with these challenges and their solutions in detail.

4 Towards a Flexible Multi-Protocol Network-Layer

4.1 Introduction

The core aim of this thesis is to design a multi-protocol network-layer. While the previous chapter introduced the idea of such a layer, this chapter continues the discussion in a more practical manner by concentrating on matters that are relevant to its design. Firstly, the proposed design of the layer is set within the context of existing network-layer designs. A basic blueprint, which is used to guide the design of the components of the multi-protocol network-layer, is then developed. The key assumptions and propositions underlying the design of the layer are briefly introduced, forming the basis on which subsequent design choices are made. Finally, the structure of the basic network-layer, which includes the three routing protocols: AODV, DSR and OLSR, some attendant data structures and the experimental Network-Layer packet are also described in this chapter.

4.2 Network-Layer Design

Before focusing on the design of a new type of network-layer, it is appropriate to look at existing network-layer design and to consider both the purpose and position of the network-layer within the communications stack in more historical detail. The traditional design of a network has consisted of an interconnected system of routers, gateways, switches and hosts,; each incorporating a rigid multi-layer communications stack which generally corresponds to the Internet's TCP/IP stack. While routers, i.e. nodes

dedicated solely to routing, do not deal with issues above the network-layer, ad hoc nodes are generally both host and router, comprising of layers from the application-layer right down to the physical-layer.

The International Standards Organisation's (ISO) Open System Interconnection (OSI) seven-layer reference model clearly defines the purposes of each layer of a theoretical communications stack with regard to *services*, *interfaces* and *protocols*. Layers are created where different levels of abstraction are required for the purposes of providing specific peer-to-peer services. The OSI reference model provides a guide to enable the development of interoperable and competing open systems; hence the need for a clearly defined stack structure. Each layer provides a specific service which higher layers may access through associated interfaces. The specific details of the protocols contained within each layer, which go about providing the layer's services to the higher layers, are not defined. Such details are left to the discretion of the protocols' designers and implementers. The stated purpose of the OSI network-layer is to control a subnet, i.e. a network of routers excluding hosts, with specific regard to the routing of packets, the control of congestion, the accounting for services provided and the interconnection of heterogeneous networks. Consequently, an OSI-type network-layer may have multiple interfaces to provide access to the multiple services that it provides.

While the model espoused by the ISO's OSI is a strong model, in that it provides for a clear distinction between the various elements of the stack, i.e. the layers, services, interfaces and protocols, the protocols that have been developed for OSI-based networks have not been widely adopted. The most popular networking technology is not based on OSI standards; rather it is based on the standards that have grown out of DARPA's ARPANET, i.e. the TCP/IP suite of protocols and the stack structure that has been built around them. DARPA wanted a flexible packet-switched networking technology that could facilitate communication between dissimilar and remote computer systems. The system had to be robust to Cold War Era threats while supporting multiple divergent applications. With the goal of heterogeneity in mind they settled on a network-layer, which they called the internet-layer, that offered a connectionless packet-switched service from a source host to a destination host across an internetwork of routers. Packet-switching means that packets are independently routed through the TCP/IP network from one end to the other. The only services that the TCP/IP internet-layer provides are the sending and receiving of packets. Unlike the OSI model, the TCP/IP model lets higher layers such as the transport-layer, i.e. TCP, deal with connection-oriented services which are based on the internet-layer service of the unreliable data packet. In providing such a packet-switching service, the routing of packets is a major issue for the TCP/IP internet-layer.

The differences between the two communications stack reference models may in someway be accounted for by the fact that the OSI model was devised prior to the development of any OSI-standard protocols such as the Intermediate System-to-Intermediate System (IS-IS) routing protocol. As such, the OSI model was not biased towards any specific protocols although it has been noted that the OSI's seven-layer model bears a striking resemblance to International Business Machine's (IBM) System Network Architecture (SNA) which IBM were promoting at the time of the OSI's development. Incidentally, IBM was the major computing corporate entity at that time.

In the case of TCP/IP, the protocols were designed first, engineered to address the problem set identified by DARPA, and the four-layer stack model was retrofitted to the protocols afterwards. TCP/IP became the dominant networking technology for many reasons, not least because it was free. TCP/IP was included in Berkeley Software Distribution (BSD) UNIX software which was widely distributed. Through such free distribution it got a foothold in the market, such as it was, and became locked-in to a near-monopoly through early accidental factors. On the other hand, the development of OSI-standard stacks suffered as the OSI was criticised for the replication of services across multiple layers and the disparity of the amount of functionality assigned to various layers. The presentation layer is wafer thin while the data link-layer has had to be subdivided into sub-layers because it was originally assigned too many services.

It has been noted by Steven Deering, a senior Cicso Systems Internet architecture researcher, in presentations that he has made to the ACM, the IETF, and others, that the TCP/IP stack has an *hourglass* shape. By that he means that a wide variety of applications and end-to-end upper layer protocols are supported by a single and narrow internet-layer protocol which depends on the support of a number of data link and physical-layer protocols. The narrow functionality of the network-layer enables more interconnectivity between heterogeneous systems of networks, as is evidenced by the present structure of the Internet. The structure of the Internet is quite formal, rigid and hierarchical, comprising of autonomous systems (ASs) that are connected by high bandwidth backbones. Within an AS, routers are organised to use a specific routing protocol, an interior gateway routing protocol such as OSPF, and routing between these homogeneous domains is done by means of exterior gateway protocols such as the Border Gateway Protocol. It is possible to create such a seemingly complex and heterogeneous networking system as the service provided by the TCP/IP internet-layer is so basic, i.e. the sending and receiving of connectionless data packets from a source host to a destination host. However, when additional services such as QoS and multicasting are added to the network-layer the complexity of interconnecting heterogeneous systems increases as such services must be replicated across the various disparate elements of the internetwork.

The traditional router design has been more than adequate for the needs of Internet system development over the last couple of decades. The protocols within each layer of the stack have been versatile and dynamic enough to react to changes in the networking environment, meeting the Cold War Era threats for which they were developed in the 1970s whilst also addressing the more recent challenges of anti-commerce and social-disruption-motivated denial-of-service attacks or terror-motivated attacks. The Internet has also been able to cope with the explosive growth in both its size and its usage; while more subnets and hosts have been added than were originally forecast, the Internet has managed to scale well thus far. Over that time, the application-layer, physical-layer and link-layer have changed as new transmission technologies have evolved and new applications have been popularised. The original applications of the ARPANET such as E-mail, Telnet (a terminal emulation programme) and FTP (a file transfer protocol) have been added to significantly with the introduction of HTTP and the world-wide-web. Changes in applications have occurred at the end-points of the network, on host machines, away from the core of the network where routing takes place. The physical-layer technologies, used to connect one node to another neighbouring node, have advanced from copper, with improved modulation schemes, to fibre-optics and satellite links. Changes in the physical-layer technologies have taken place locally between neighbouring machines. There is a relative ease in the upgrading or changing of such layers of the communications stack when compared to the

network-layer as neither the end-to-end application-layers nor the local physical-layer links have a global significance for the operation of the internetwork. If a new application-layer is introduced to a number of host machines, the rest of the network will continue to operate, albeit in the presence of increased traffic and increased demand for the services of lower layers.

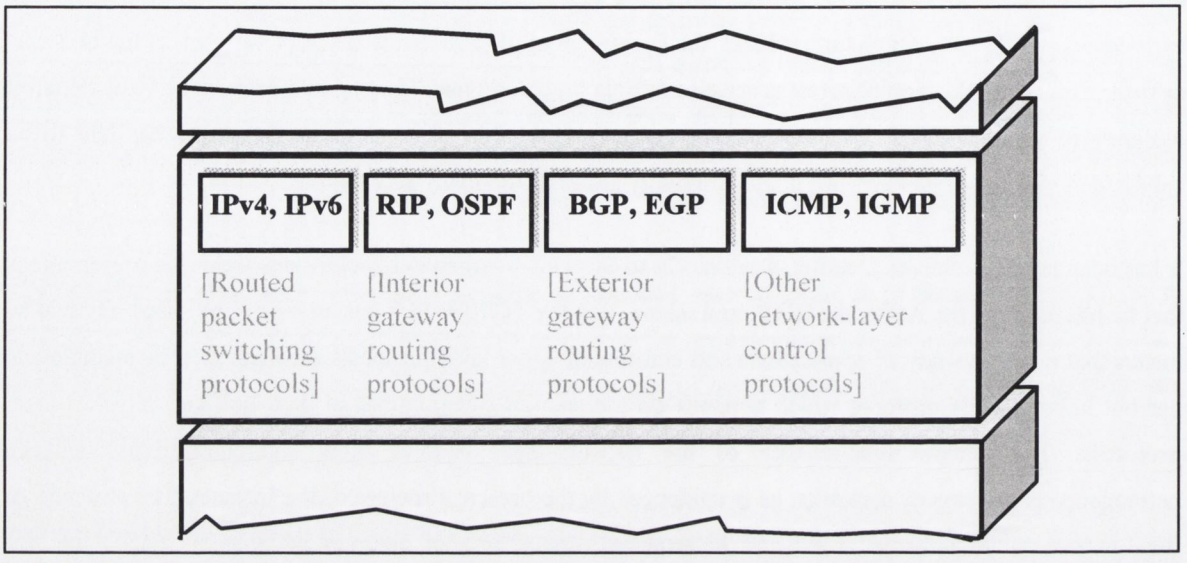


Figure 4-1 TCP/IP Internet layer

A notable point of difference between ad hoc networks and traditional static networks is that the intelligence of traditional wired networks, such as the Internet, is in the routers at the centre of the network. As such, most routing protocols such as RIP and OSPF are router intelligent rather than host-intelligent as DSR is. The host machines are generally quite dumb at the network-layer; their only interaction with the network-layer, as regards packet routing, is to present a packet to a pre-assigned router which decides which way to route the packet to the stated destination and to receive packets from the router. Traditional networks have also had longevity; they have been set-up to last for years and the networks have changed at a different pace to the hosts which have seen rapid changes in the types of applications and services running on them. To routers, change at other layers appears as either an improved underlying service, i.e. more stable or high bandwidth underlying links, or as increased demand for the network-layer's service, i.e. more traffic from higher layers.

However, the pivotal nature of the network-layer, as the layer that enables the creation of a flexible and heterogeneous internetwork, also inhibits the manner in which the network-layer can be changed or reconfigured. Figure 4.1 depicts some of the array of protocols that may currently be deployed in the internet-layer. As illustrated, there are multiple interior routing protocol options, addressing options and exterior routing protocol options. Most routers are equipped with a suite of such protocols. Routing protocols and packet-switching protocols need to be changed when the characteristics of the network change. Traditional networks may be characterised in terms of the bandwidth of links used, the number of hosts in the network and the number of expected hops within the AS, among other factors. When these factors change, new solutions are demanded at the network-layer. Since the network-layer is the glue that binds disparate routers and networks together, the manner in which change occurs at the network-layer is very protracted as it requires detailed planning and co-ordination. It is also subject to many different political, technical and commercial interests. For example, in response to the explosive growth of the Internet and the expected

exhaustion of the IPv4 address space, the IETF have proposed the IPv6 standard as a replacement technology at the network-layer. The adoption of IPv6 is seen as a long-term solution that should eradicate the need for other short-term and sometimes ad hoc measures such as Classless Inter-Domain Routing (CIDR) and Network Address Translation (NAT). IPv6 is not compatible with IPv4, but will work with slightly modified versions of other network-layer protocols such as ICMP, OSPF, EGP and BGP. In order to ease the migration from IPv4 to IPv6, most routers now offer a dual operation approach which allows for the transition to occur in a non-traumatic manner. Tunnelling techniques are also used as a short-term measure to enable IPv6 traffic to cross IPv4 regions of routers. In terms of routing protocols, while OSPF was standardised as an upgrade replacement for RIP in 1990, the migration process is still ongoing. OSPF was developed to address the perceived shortcomings of RIP as RIP was not designed for the characteristics of ASs that exist today, i.e. ones which have a larger number of routers, routes greater than 16 hops and routes comprised of links of varying capacity. To this end, a standard Cisco router now incorporates standard and legacy routing protocols to enable smooth migrations within networking domains. So, while the characteristics and network-layer requirements of such networks change over years, the technical solutions that are provided at the network-layer also take years, if not a decade, to be realised due to the critical nature of the network-layer. In marked contrast, and as noted throughout this thesis, the pace of change of mobile ad hoc networking characteristics and the diversity of mobile ad hoc networking environments are much different to those experienced in the traditional wired static networks.

4.3 Mobile Ad Hoc Network-Layer Design

Mobile wireless ad hoc networks are the focus of this thesis and they are radically different to the traditional, wired, static networks that have existed heretofore, notwithstanding the fact that the network-layer of an ad hoc network router is still charged with providing an equivalent unreliable packet routing service. The range of new routing protocols have been developed for incorporation into the network-layer of an ad hoc node as OSPF or RIP were inserted into the TCP/IP stacks of static routers. Protocols have been developed using reactive, proactive and hybridising approaches. Several ad hoc routing protocols are being brought through an open standardisation process which suggests the possibility of there being a number of standardised generic ad hoc routing protocols in the near future.

While routing protocols for ad hoc networks are an improvement on the likes of IS-IS, OSPF and RIP, in that they are more bandwidth conscious and react to topology changes faster, they are also constrained by an inability to react to changes in the networking environment beyond the topology changes for which they were designed. The protocols take no account of the conditions of the network in which they operate. The literature suggests that the typical stack of an ad hoc node will have the form illustrated in Figure 4.2. The routing protocol, whether it is AODV, DSR, DSDV, OLSR or some other protocol, will be statically deployed as the routing protocol component of the mobile ad hoc network-layer. Indeed, other networking issues such as the interconnection of heterogeneous ad hoc networks are still discussed and developed in terms of the interconnection of ASs by means of fixed gateways and interior and exterior routing protocols.

It would appear that this derivative approach does not fully acknowledge that mobile wireless ad hoc networks, by virtue of their inherent attributes, necessitate a sea-change in the way that networks, by means

of the network-layer, are formed and managed. Every node in an ad hoc network is also a router; network *intelligence* is distributed among all of the nodes and is no longer the preserve of a centralised core of static routers, switches and gateways as it is in older traditional networks. Due to the inherently distributed nature of an ad hoc network, all nodes must share responsibility for the provision of the network-layer service so that higher layer multi-hop peer-to-peer services may operate. While mobile ad hoc networks are self-forming, their existence is also transient in nature as they are built on-demand. As such, the structures and concepts associated with the Internet such as hierarchically managed ASs and gateways are not viable in a mobile ad hoc networking environment.

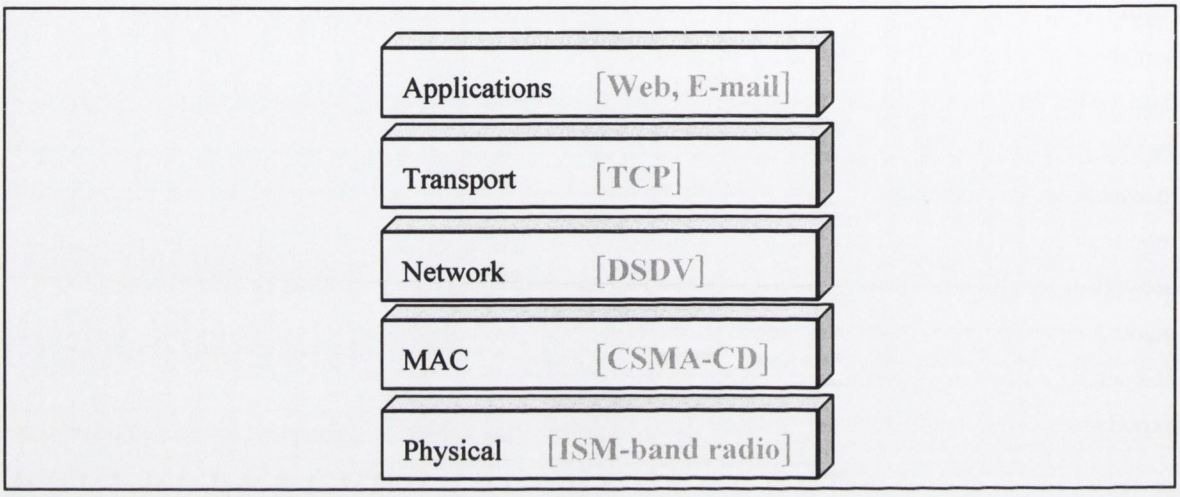


Figure 4-2 Typically Suggested Structure of an Ad Hoc Router Node

The literature has emphasised that mobile ad hoc networks will be challenged by a variety of stressful networking scenarios, characterised in the main by varying node densities, node mobilities and traffic loadings. During the process of designing, reviewing, evaluating and modifying the proposed ad hoc routing protocols it has been shown that no single ad hoc routing protocol is suited to the entire range of likely networking scenarios. In essence, it has been shown that the network-layer needs to use different routing techniques to deal with diverse networking scenarios. It has also been noted that the concurrent IETF standardisation of multiple unicast routing protocols, such as AODV, DSR and OLSR, will result in the availability of a number of mature open routing protocols that are suited to different networking dynamics. As mobile ad hoc networks can be built in the absence of centralised support structures provided by Telcos, the development of multiple ad hoc networks using a combination of such open routing standards is very likely. The notion of using single-protocol nodes, as depicted in Figure 4.3, to construct robust and realistic mobile ad hoc networks suited to diverse scenarios and varying conditions is questionable. Such inflexible single-protocol ad hoc routers would be restricted to specific networking scenarios and networking conditions. For example, the node depicted in Figure 4.3(b), operating DSR, would be suited to a low density network experiencing long-lived traffic but would fail in a dense network experiencing sporadic traffic, unlike the node depicted in Figure 4.3(c) which operates OLSR.

As Deering has noted, the existing simplicity of the service provided by the network-layer lends itself to the construction of a heterogeneous Internet, where the heterogeneity manifests itself from AS to AS; this same observation holds true for the mobile ad hoc networking environment. The network-layer still remains the

glue of any network, no matter what form its design takes. Nodes must be able to exchange topology information using a mutually understood protocol and must forward data to and from other nodes. Therefore, the proposed network-layer should still be based around the basic service of the unreliable data packet, but the manner in which the network-layer components go about providing that service should be more in tune with the reality of the mobile ad hoc networking context.

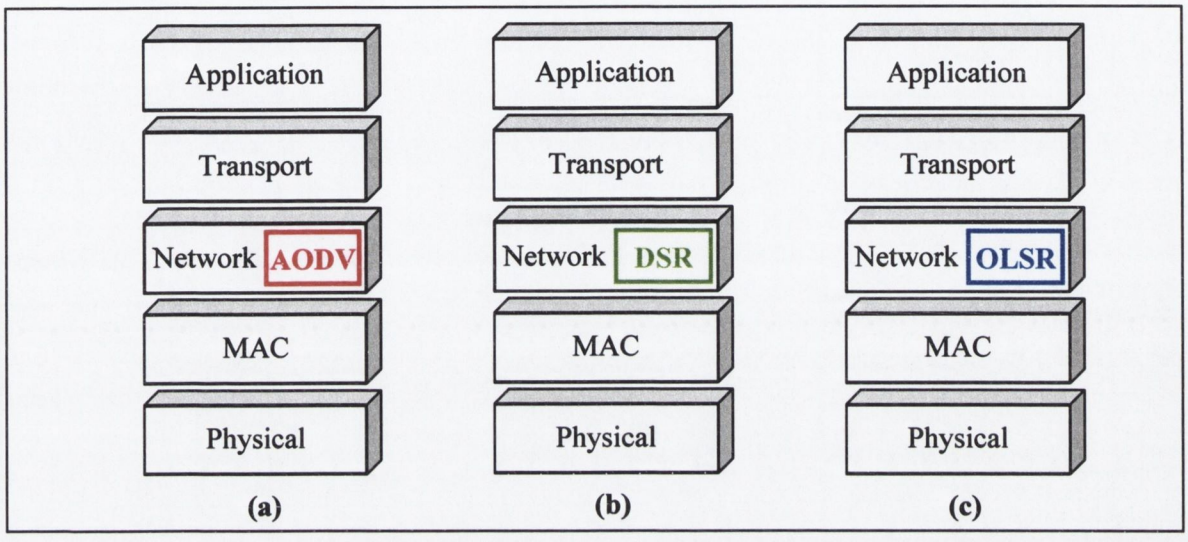


Figure 4-3 Three Examples of Ad Hoc Routers Suggested in the Literature; AODV Router (a), DSR Router (b) and OLSR Router (c)

4.4 Future Mobile Ad Hoc Network-Layer Design

Moving on from the existing static ad hoc network-layer approaches to a design suitable for a flexible network that is completely open, distributed and of unbounded size, requires that a network-layer of the type described in the last chapter is developed.

Recapping the main points of that chapter, the thrust of the idea behind this design approach is that within the space occupied by the whole network, groups of nodes will be able to organise themselves on a regional basis in response to the perceived changes in the networking conditions. The objective of a flexible multi-protocol network-layer is to allow groups of nodes, which are collaborating to form a network, to choose the protocol most suitable for the region of the network that they occupy at that time. The flexible network-layer should allow the entire network to function as a patchwork of heterogeneously configured regions so that nodes across the network expanse may still communicate on an end-to-end basis. Such heterogeneous network-wide configuration at the network-layer should not restrict the innate ad hoc network characteristic of autonomous node movement so that nodes may freely move from one region to another. The flexible multi-protocol network-layer addresses these issues of auto-configuration, internetworking and reconfiguration.

A number of general observations can be made about the system design necessary to facilitate such flexibility. Each of the nodes of the network is engaged in response to changes in the composition of the network state or to changes in the underlying networking conditions. Generally, each node should be capable of observing and analysing network events before deciding, either independently or in collaboration with other nodes, on a course of action which it must then implement.

As the ad hoc routing protocol is a dependent variable of such a flexible ad hoc networking system, each node is tasked with ensuring that a consistent network state, i.e. the use of a common routing protocol, is brought about in the multimodal network by the participating nodes. The global state of the network at any given time may be viewed as the union of the states of each of the participating nodes and the messages that are in flight at that time.

In order to realise a consistent and optimal global state in the network, nodes should be enabled to observe and analyse the network. Observations and analyses take two forms. The first type of observation and analysis is that of the prevailing networking conditions. Nodes should be able to appreciate the networking conditions in which they operate, e.g. node mobility rates, node density and traffic, as such factors have been found to strongly influence the efficiency of ad hoc routing protocols.

The second type of observation and analysis is that of the composition of the network. A flexible system, such as the one proposed here, allows nodes to operate different protocols as demand arises. This may possibly lead to a heterogeneous network composition from the point-of-view of multiple network-layer configurations. Nodes should be able to ascertain the composition of the network, in so far as it affects them, so that they can take informed actions to rectify the global network state.

In arriving at an optimal and correct global state, a node may make a number of different decisions which may result in a change of state at the local node. One type of decision may be reactive and independent in nature, taken in response to observed inconsistencies in the network state. Such a decision does not require collaboration or consultation with other nodes. This type of situation may typically arise during node migration or network mergers. However, if a node seeks to change the existing valid state of the network, i.e. change the routing protocol, in response to observed changes in the networking conditions, such a decision should be taken in consultation with the other affected and dependent nodes as such a decision is dependent on the will and ability of those nodes to participate in the desired modification of the global state of the network.

The general functionality of the flexible multi-protocol network-layer, which is realised by the aforementioned observation, analysis and decision-making features of each node, is summarised in the schematic depicted in Figure 4.4. As can be seen in Figure 4.4, the flexible multi-protocol network-layer consists of a suite of distinct ad hoc routing protocols, each with its own performance strengths and weaknesses, and a number of components that enable the nodes to either unilaterally, or multilaterally, configure the choice of ad hoc routing protocol for each node's network-layer. The particular protocols used for this body of research are described in Appendix A. The flexible network-layer adapts its configuration in response to perceived changes in the network, whether they are changes in the configuration of neighbouring nodes or changes in the networking conditions. The flexible multi-protocol network-layer uses observations about its current state and about the network to propose an action (i.e. to auto-configure, to reconfigure or to internetwork). The manner in which the network-layer makes observations and the types of observations that are possible are topics that are examined in Chapter 5. As previously stated, a node may take unilateral actions, i.e. it may configure itself based only on local knowledge, or more complexly, the node may propose actions that require multilateral feedback before it executes a reconfiguration of the network-layer. The

mechanisms used for making unilateral and multilateral decisions are the focus for Chapters 7 and 8. Once the decision is made, the action to change protocol is executed.

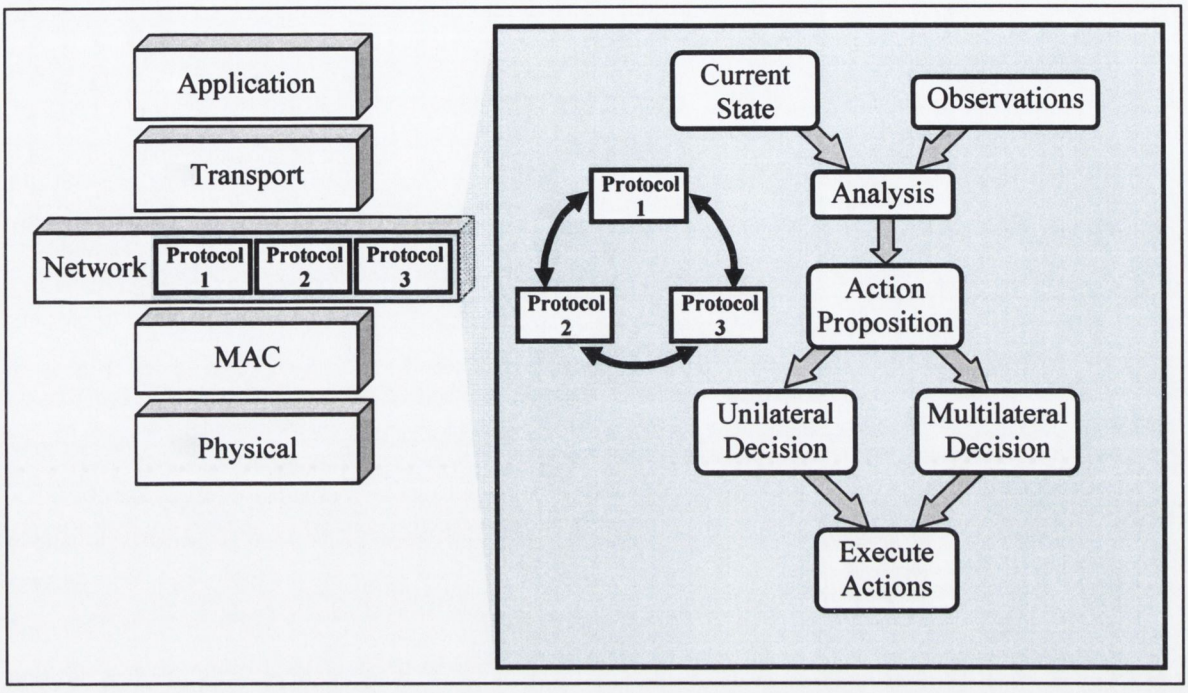


Figure 4-4 A Basic Blueprint to Guide the Flexible Component Design

The flexible network-layer can therefore be said to consist of a number of components that are local to a node and facilitate observation, analysis and decision-making in a distributed entity with the attributes of an ad hoc network. A component is a loose term used to delimit functionality within the system; each component employs a set of techniques that fulfil certain distinct system requirements as laid out in this chapter and in Chapter 3. For the sake of clarity, it has been found helpful to present this work in terms of a composite system based on the development of each of these flexible components. As stated already, the remainder of the chapters in this thesis focus on the design of these components. Figure 4.4 acts as a blueprint in working through the design.

4.5 Design Restrictions

There are many design decisions to be made in creating the components of the proposed flexible network-layer. Those design choices can only be made by taking into account some assumptions and propositions about the type of ad hoc networking system within which the layer will operate. These assumptions and propositions have a major impact on the design choices and design implementation and will be taken as given in all subsequent discussions.

4.5.1 Assumptions

The primary assumptions regarding the ad hoc network for which the network-layer is designed are listed here and reasoned in the succeeding paragraphs.

- The system is assumed to be independent of external systems, services and pre-existing infrastructure.
- The system is assumed to lack accurate time synchronisation.
- The system is assumed be open, cooperative and secure.

■ Independence of External Systems, Services and Pre-existing Infrastructure

It is assumed that the network does not have access to any external systems, services or pre-existing infrastructure. The reliance on external technologies such as the U.S. Department of Defence's GPS for position information or the PSTN for out-of band signalling introduces a potentially critical dependency into the ad hoc network. These external infrastructures may place technical, financial, social or political barriers in the way of the successful operation of an independent ad hoc system.

Position information is at present available from a system such as the GPS. The E.U.-led Galileo project may provide such services to future ad hoc networking systems, but possibly at a financial cost, unlike the GPS. As noted earlier, some routing protocols [Ger00], [Bas98], rely on location-positioning information to enable their multi-hop routing mechanisms. In recognition of the potential problems resulting from an external system dependency, some have sought to determine position information without recourse to external services. In [Cap01], the authors have proposed a scheme that provides nodes with relative position information based on their position within the network without the use of GPS services. The scheme is a response to the authors' quest to design a distributed and infrastructure-free positioning algorithm that would not compromise the robustness of the ad hoc network.

The operators of the PSTN may enforce strict regulations regarding the types and sources of traffic that use their network. They may also impose their own authentication, authorisation and accounting practices. Other difficulties may arise if a section of the PSTN on which the ad hoc system is reliant fails or if the GSM system goes offline for scheduled maintenance. The solar storms of autumn 2003 resulted in the temporary loss of GPS services over large swathes of the northern hemisphere. It is the author's contention that for an ad hoc network to be designed as a replacement technology that fills the void brought about by failing conventional technologies, whether due to man-made or natural disasters, then it should not be tethered to those systems.

■ Absence of Accurate Time Synchronisation

It is assumed that the global synchronisation of nodes' processor clocks is not feasible. Nodes may exist on a variety of platforms with varying degrees of accuracy and clock drift. The synchronisation of clocks across a wide network diameter with variable multi-hop routing capabilities and transmission media is not trivial [Cou01]. Lacking an external beacon for clock synchronisation, such as GPS or a base-station, it holds that the nodes of the network will most probably be adrift in time. The Network Time Protocol [Mil91] is often used in fixed networks to provide time information to remote routers and hosts so that they may synchronise events. The use of such a protocol does not meet with the other declared assumptions and propositions; namely that a dependence on centralised or external systems is eschewed.

■ **Open, Cooperative and Secure**

It is assumed that the network is an open, cooperative and secure one. An open network allows nodes to join it or leave it without seeking permission from any supervising agency. In an open network, or indeed in a closed military or commercial network that is sensitive to intrusion and attack, the issues of security and cooperation are quite important. As an example, Zhou *et al.* [Zho99] note that some of the challenges that must be surmounted to provide security in ad hoc networks include the dependence of the network on wireless links that are susceptible to passive eavesdropping, reply attacks and message distortion. These events require that trust should be established dynamically between varying subsets of interacting nodes. The authors have proposed a fully distributed scheme that relies on threshold cryptography to provide a highly secure and highly available key management service which addresses those attributes. Others like [Zha00], noting the vulnerability of mobile wireless ad hoc networks to attack and the pressure that this may bring to bear on distributed decision making processes, have proposed a scheme that allows for the detection of intruders in a wireless ad hoc network. The introduction of adversarial intruders into an ad hoc networking system that relies on distributed cooperation may lead to Byzantine failure in that system [Lam82]. In this thesis it is assumed that Byzantine failures do not occur.

While an ad hoc network may be secured and guarded against attack in a distributed manner, such security services do not provide an incentive for nodes to cooperate when they operate under their own steam. In a military or rescue mission operation there is a defined hierarchy that enforces cooperation between the constituent nodes of the network. Nodes will not act in a selfish manner as they are tasked with a common goal and operate under a single authority. In an open network where each node may be its own authority, there is no social or hierarchical impetus to provide services like multi-hop routing to other adjacent nodes. This problem has been addressed in part by the Terminodes project [Bla01] which has investigated a range of issues concerning the cooperation of nodes in self-organising ad hoc networks. Both [But00] and [But03] describe schemes to stimulate and enforce cooperation between nodes. In particular, [But03] presents a protocol that has been designed to stimulate nodes to forward packets for other nodes, to discourage nodes from abusing network resources, e.g. by overloading the network with data packets, and to stay alive even when the node itself does not need to access networked services so that it may be of use to other nodes. The schemes complement the aforementioned security schemes of [Zho99] and [Zha00] that guard against malice by incentivising cooperative behaviour among secure nodes. In short, the assumption that an autonomous civilian ad hoc network may be open, cooperative and secure is not beyond the bounds of current security and cooperation schemes.

4.5.2 Propositions

Based on the discussion in the previous section, two further restrictions are added in the form of propositions. These are listed here and reasoned in the following paragraphs.

- It is proposed that the system be completely distributed in design.
- It is proposed that the system design be compatible with, and sympathetic to, the network-layer fundamentals.

■ **Distributed in Design**

It is proposed that the network-layer should be completely distributed in design. Hierarchical and centralised design approaches may place particularly stressful demands on a mobile wireless ad hoc network's operation [Per01]. The nodes at the higher tier of a hierarchy must be selected or elected in some way. They may be either manually selected or a cluster-based algorithm may be used to dynamically elect the nodes forming the upper tier of the hierarchy. Clustering algorithms have been of interest to researchers in the ad hoc networking community as they provide virtual structures that enable many types of efficient control schemes [Che04]. They have been used for radio transmission management when the reuse of the channel is required, e.g. for time or frequency division schemes. They are also used to set-up backbones which are used to provide low-delay, high-speed connectivity between distant nodes in large networks [Xu02]. Some multi-hop routing schemes have also been designed around the notion of clusters. Cluster-based control structures require algorithms for initially clustering nodes and for maintaining clusters as nodes join, leave or move within the network; essentially requiring that state information akin to route state information be maintained throughout the network. Clustering algorithms are generally targeted to quasi-static networks.

However, a naturally occurring ad hoc network event such as network partitioning has a detrimental effect on hierarchical networking solutions if nodes from the upper-tier of a hierarchy become separated or isolated from the lower-tier nodes. The schemes must adapt to such events by electing new leaders or cluster heads [Sun03b]. Upper-tier nodes will also consume more resources through their increased message processing and packet transmissions. In an open networking environment, one in which there is no natural social hierarchy, nodes may be unwilling to assume the cost of being in a managing or co-ordinating role. In this thesis, an effort is made to avoid any centralised or hierarchical approaches that may introduce significant points of network failure.

■ **Compatible with Network-Layer Fundamentals in Design**

It is proposed that the design of the flexible system should be compatible with the fundamental properties of the underlying network-layer elements. The fundamental elements of the network-layer are the ad hoc routing protocol in use and the data forwarding protocols employed. The core protocols used in an ad hoc network-layer may vary from reactive protocols such as DSR, AODV and TORA to proactive protocols such as DSDV and OLSR. This thesis is only concerned with distributed unicast ad hoc routing protocols.

The ad hoc routing protocols have been designed to minimise control overhead, specifically the number and size of transmitted control packets, in a trade-off with efficiency of operation. Protocols such as DSR and AODV are reactive by design; they throttle back the amount of overhead incurred when demand for routing services is low. Protocols such as OLSR, on the other hand, use periodic control message transmissions to gain up-to-date network topology knowledge regardless of the demand for routing services. However, as a multi-protocol system is proposed which should allow the routing protocols to be changed in real-time, the signalling characteristics of the system will change. Accordingly, the system should be designed in such a manner that it reflects the changes in the fundamental network-layer elements.

A compatible design is one that is sympathetic to the underlying design of the core protocols, rather than working against them and detracting from their desirable traits which have been tailored to their operating environment. For example, it is argued that the use of extensive proactive signalling in the course of use of a reactive protocol would defeat the purpose of using a reactive protocol that throttles back its signalling overhead when demand for its network-layer service falls off.

It is also proposed that the system should not introduce any cyclic dependencies. The purpose of the system is to allow a multi-protocol network-layer to respond to the network needs in a flexible manner. The ability of such a network-layer system to operate, i.e. to reconfigure the network-layer of nodes in response to changed conditions or network events, should not be dependent on the availability of the multi-hop routing services themselves.

4.6 The Basic Network-Layer

This section briefly introduces the basic components of the flexible multi-protocol network-layer. The flexible multi-protocol network-layer sits at layer 3 of an OSI-type stack, i.e. the network-layer, or at layer 2 of a TCP/IP-type stack, i.e. the internet-layer. The layer provides a routing service analogous to the more traditional TCP/IP internet-layers. Specifically, the proposed network-layer accepts requests from higher layers to route data to specific unicast destination nodes. The proposed network-layer provides an unreliable packet-switching service. Connection-oriented services are left to the discretion of higher layers which may use the connectionless services provided by the proposed network-layer.

The flexible multi-protocol network-layer is comprised of a number of distinct components which can be divided in two categories; *primary routing* components and *secondary flexibility* components. A network-layer comprised of the primary routing components operates in the same manner as the statically configured nodes cited in the general ad hoc networking literature; such a node can only operate one of the available protocols which is chosen offline before the node boots-up. The primary routing components that form the basic network-layer are listed below:

- Ad Hoc On-Demand Distance-Vector (AODV) protocol
- Dynamic Source Routing (DSR) protocol
- Optimised Link-State Routing (OLSR) protocol
- Route Cache
- Neighbourhood Table
- IPv4-type Packet Switcher
- Other common supporting data structures and state variables for routing components

4.6.1 Routing Protocols

The AODV, DSR and OLSR protocols have been used for this thesis in order to tease out and develop a multi-protocol network-layer that has routing components suited to a wide range of networking conditions. The operation of each protocol is sufficiently different to warrant that a clear description of each protocol is provided in Appendix A for those who are unfamiliar with the three protocols. As the protocols use different

routing mechanisms, they demand that the techniques used in the system are as generic and widely applicable as possible, addressing the needs of source-routed reactive protocols (DSR), distributed reactive protocols (AODV) and time-triggered proactive protocols (OLSR). Other ad hoc routing protocols could be used, in this work, in place of the protocols chosen. Indeed, a location-based protocol, i.e. one using a GPS-like system, could also have been used if the weakening of the design criteria discussed in the last section was acceptable.

The protocols have been independently developed in accordance with the IETF specification documents for each protocol [Per02b], [Joh02b], and [Cla02]. They may be run independently, as statically deployed routers using the basic network-layer structure, or as part of the flexible multi-protocol network-layer. Each protocol is uniquely identified in the various packet headers in the proposed system by means of unique prime number identifiers; AODV: 2, DSR: 3 and OLSR 5.

4.6.2 Route Cache

Each of the routing protocols uses a route cache of unique structure. AODV uses a cache based on distance-vector information. DSR uses a cache based on source-routes. OLSR uses a cache based on link-state information. In addition to the cache, OLSR maintains other topology information tables such as a one-hop and two-hop neighbourhood cache. The route caches are managed according to the specifications for each respective protocol.

To illustrate the different structures used by the three candidate protocols, Figure 4.6 depicts the route cache entries of Node 1 corresponding to the sample network consisting of ten nodes and two traffic streams, the dashed red lines. Node 1 is forwarding data to Nodes 9 and 11 along the paths indicated in the diagram. Figure 4.6 depicts the three possible route cache structure of Node 1 if it was configured to use either AODV or DSR or OLSR. The distinct route information that is recorded by each protocol is clearly evident in the diagram. OLSR has full state knowledge, regardless of the traffic demands, whereas the AODV and DSR route caches reflect the localised and demand-driven nature of the routing information held by nodes operating those protocols. The expiry times for the entries of each of the three route caches, which are not shown, are set by the routing protocols and differ according to their route discovery and route maintenance procedures, as discussed earlier in this chapter.

Also, while the DSR protocol specifically elaborates on the source-routed packet-switching technique to be used in a DSR network, AODV and OLSR specify that packet-switching should take place in accordance with the IPv4 Router specification, IETF RFC 1812 [Bak95]. A simple packet-switching component that effects the basic packet-forwarding scheme described in [Bak95] is provided in the network-layer for AODV and OLSR nodes.

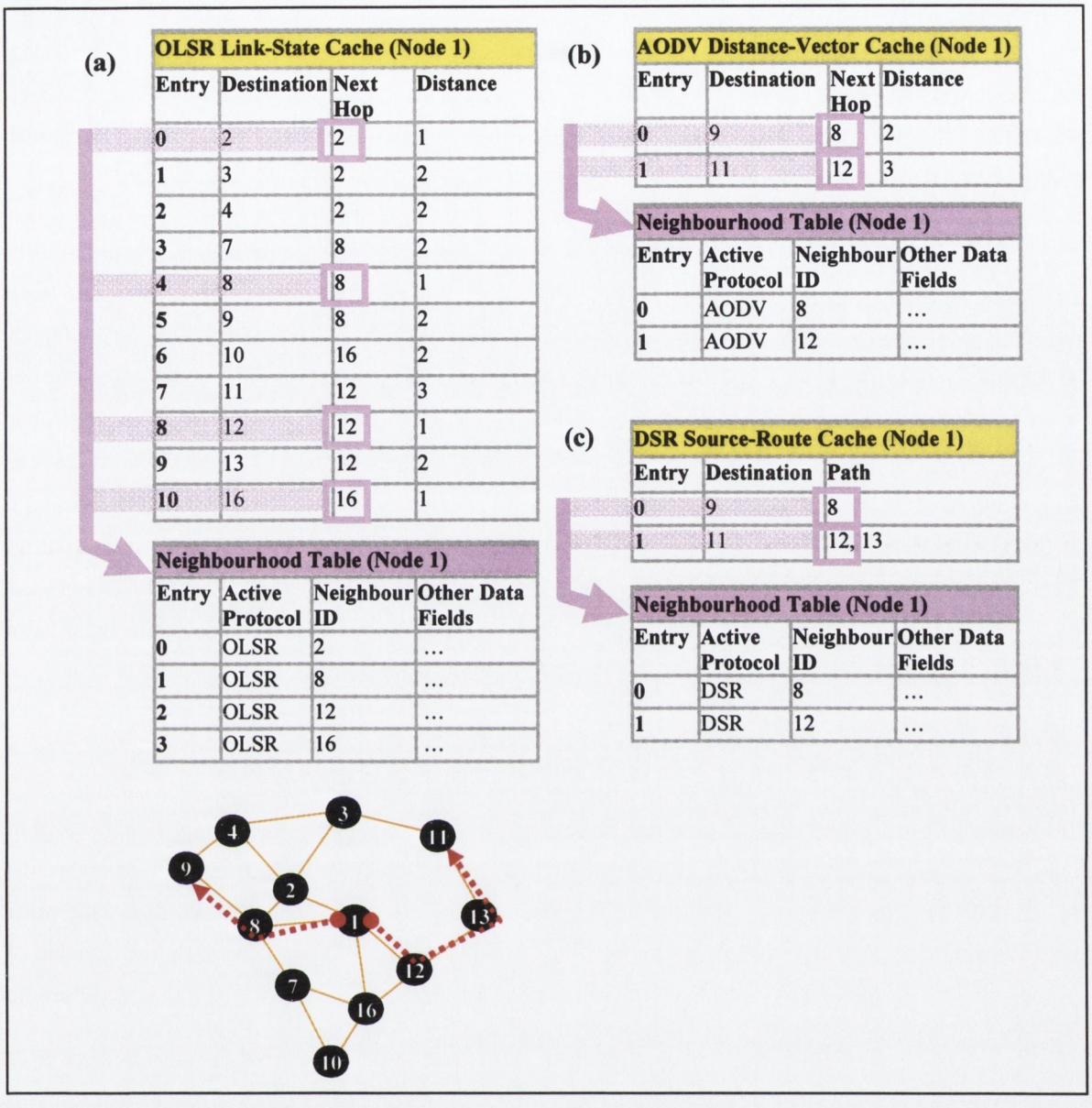


Figure 4-5 Three Distinct Route Cache Structures and Associated Neighbourhood Table Entries

4.6.3 Neighbourhood Table

The Neighbourhood Table plays an important role in the proposed flexible multi-protocol network-layer. The Neighbourhood Table is a virtual data structure based on elements of the three candidate routing protocols' existing data structures. While the Neighbourhood Table is not necessary for the operation of standard pre-configured networks, it is used extensively in the flexible multi-protocol network layer. It is introduced here and is described in much greater detail in the next four chapters as it is used.

Of the three candidate protocols, only OLSR explicitly maintains a *Neighbour Set* as part of its topology discovery mechanism. Through the issue and receipt of Hello messages, the OLSR protocol tracks its neighbouring nodes as links come-up or go-down. While neither AODV nor DSR explicitly maintain a Neighbourhood Table, they do implicitly record one-hop topology information in their route cache structures. As noted earlier in this chapter, AODV and DSR do not ordinarily actively monitor their neighbouring environment; although AODV does monitor links on active routes, i.e. established routes which are in use.

During periods when AODV and DSR nodes are actively signalling, i.e. during the route discovery, route maintenance and packet forwarding phases, it is possible for nodes to learn of their neighbouring environment. Nodes may note the existence of new neighbours or the loss of older neighbours. The effects of the varying levels of signalling activity on a node's ability to sense its neighbourhood are explored throughout Chapter 5 and in Section 6.3.

The proposed virtual Neighbourhood Table structure is common to all candidate protocols. By virtual, it is meant that the existing OLSR Neighbour Set and AODV and DSR Route Caches are left as they have been specified, but certain entries in each of these structures are treated as entries in the Neighbourhood Table. All of the one-hop neighbour entries in the OLSR Neighbour Set and the one-hop route cache entries in the AODV and DSR route caches are treated as entries in the Neighbourhood Table. The basic entries in the Neighbourhood Table simply record the identities of existing neighbouring nodes, as shown in Figure 4.6. Entries appear in the Neighbourhood Table when they have been added to the underlying routing protocol's cache or neighbour set; they are removed from the Neighbourhood Table when the underlying routing protocol expunges them from its cache or neighbour set. The basic neighbour entries in the Neighbourhood Table may be extended to contain other data fields, which may record more detailed information about these neighbours; these extensions are motivated and described in Chapters 6 and 7.

4.6.4 Encapsulating Network-Layer Packet

The network-layer described in this thesis is part of an experimental packet-switched system which is used to form standalone networks that do not rely on pre-existing networks or infrastructure. Therefore, the encapsulating network-layer packet used in the system does not have to be compatible with other encapsulating network-layer packets such as the IPv4, IPv6 or IPX packet formats. As such, the structure of the header of a simplified encapsulating network-layer packet has been designed ab initio as there are no legacy or interoperability issues to be accounted for in the structure of this packet.

The Network-Layer header, as shown in Figure 4.7, is conceptually divided into end-to-end fields and node-to-node fields. End-to-end fields are fields that have relevance across the chain of nodes from source to destination, whether the packet is broadcast or unicast. They are initialised by the source node, but may be repeated by intermediate nodes which may modify state information regarding the journey of the encapsulated packet. On the other hand, node-to-node fields are those which have local scope only. These fields are set by a node when it transmits the packet, regardless of whether the node is originating or forwarding the packet.

The **TTL** and **Hop Count** fields are complementary header fields. The TTL field is initially set by the ad hoc routing protocol of the originating node. The TTL value is decremented by any node that receives the packet. The TTL limits the life-span of the packet within the network as the packet cannot be forwarded when the TTL field reaches zero. Protocols that do not scale well may set a small initial TTL value. The Hop Count of the packet records the number of hops over which the packet has already travelled. Both the TTL field and the Hop Count of the Network-Layer header are used to enable seamless internetworking in heterogeneous ad

hoc networks, as will be discussed in Chapter 8. Both the Hop Count field and the TTL field are one byte each.

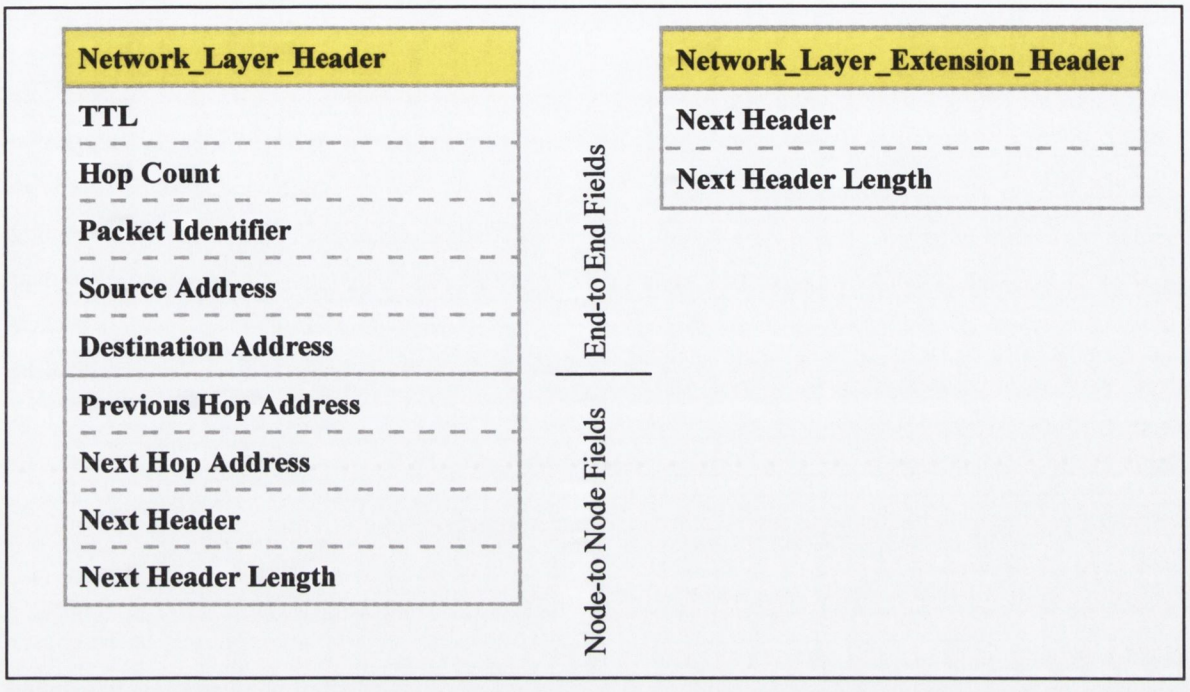


Figure 4-6 Network-Layer Packet Header

The **Packet Identifier** field is used to uniquely stamp each packet that enters the network. All nodes, regardless of their network-layer configuration may detect duplicate messages by keeping a record of the most recently received network-layer packets. The combination of the Source Address and the Packet Identifier uniquely identifies each packer in the network. The Packet Identifier field is four bytes long.

There are four address fields in the proposed Network-Layer header, as depicted in Figure 4.7; **Destination Address**, **Source Address**, **Previous-Hop Address** and **Next-Hop Address**. The end-to-end Source Address and Destination Address fields of the header are set by the originating, or source, node. This is the node that first creates the header to encapsulate other control or data packets that have their origin at this node. The Destination Address is also set by the source node and it may be a unicast address or a broadcast address. Neither the Source Address nor the Destination Address are modified as the packet is unicasted or broadcasted through the network; the values are set on an end-to-end basis, i.e. the address values pertain to the end nodes of the end-to-end transaction, whether it is a route query or data packet being forwarded.

The Previous-Hop Address and the Next-Hop Address are set on a hop-by-hop basis. The Previous-Hop Address field always records the address of the node that transmits the encapsulated packet, regardless of whether it is the originating node or an intermediary node. The Next-Hop Address field records the address of the intermediate destination of the encapsulated packet. If a packet is being flooded through the network by repeated broadcasting, then the Next-Hop Address value will always be the broadcast address identifier. On the other hand, if the encapsulated packet is being unicast through the network, then Next-Hop Address field will contain the specific address of the next node for which the packet is destined.

Each node is assumed to have a unique address. The means by which the address is assigned to the node is not at issue in this thesis. As will be described in Chapter 6, this subject is a matter of ongoing research. In the experimental system, described in this thesis, a non-hierarchical 32-bit address system is used. The choice of a 32-bit address space is arbitrary.

IPv6 employs the concept of extension headers so that the base IPv6 header need only contain core information that is frequently used by routers. If more complex router-specific or traffic-specific information needs to be the header, from one node to another, then an extension header may be added to the base IPv6 header. A similar approach is proposed in this system. All nodes, regardless of their choice of routing protocol, are expected to fully process the base Network-Layer header as it contains information about their neighbouring nodes. Further headers may be appended to the Network-Layer header, e.g. control packets such as `DSR_Route_Requests` or data packets en route, and are only processed if it is appropriate for the receiving node to do so. Hop-by-hop extensions may also be added to the Network-Layer header, such as Node Status extension which will be described in Section 6.5.

4.7 Design Platform

Before concluding this chapter there is one remaining topic that must be addressed and this is the chosen platform for the design of the flexible multi-protocol network-layer. The platform in question is the Dublin Ad hoc Wireless Network (DAWN) [O'Ma02], [O'Ma02b]. The purpose of this section is to briefly describe DAWN.

DAWN is a real ad hoc network testbed that facilitates experimentation with ad hoc networks on all levels from the application layer to the physical layer (e.g. at the security layer, routing layer, MAC layer etc.). At the core of the testbed is a dynamic modular communication stack that runs on each of the nodes of the ad hoc network. *Layers* of the stack can be independently designed in a standalone fashion. A generic layer interface allows the dynamic assembly of these layers to form a network communication stack consisting of the relevant hardware and software elements. The inter-layer interface is very simple, consisting of primitives to send information upwards or downwards through the stack. DAWN uses real mobile nodes communicating over radios that are susceptible to interference, loss of power and communications isolation and therefore offers a unique opportunity to prototype and test research ideas.

Figure 4.7 shows an example of the DAWN communication stack that would typically run on a node of the ad hoc network. As can be see from the diagram, the communication stack is broken into two parts, a base stack and an application stack. The base stack consists of components that facilitate dynamic ad hoc network communication. The application stacks attach to the base stack via connection layers. Figure 4.7 shows a number of generic application stacks running on a base stack. Typical applications that have been designed to data are chat, instant messaging and phone applications [Gre04]. The application stacks are designed in a structured manner whereby the GUI is typically separated from underlying details of the application. At the bottom of the each application stack is a layer that allows connection to the base stack. At the very top of the base stack are corresponding attachment layers for the various applications that run over the stack. A 'down

mux' layer deals with directing the appropriate incoming packet (packets coming up the stack) to the correct application.

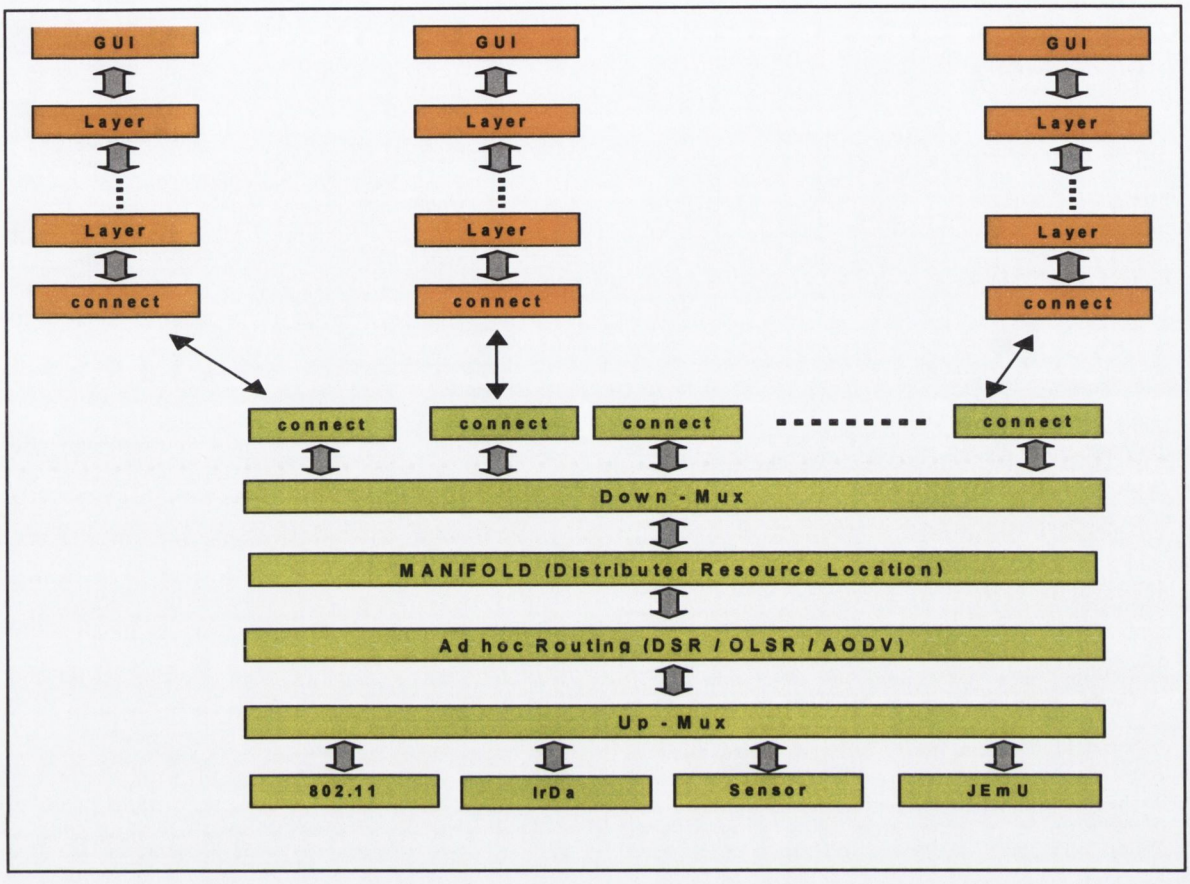


Figure 4.7 Design Platform Communication Stack Organisation

The base stack has a number of core layers that provide the type of functionality needed in most ad hoc network applications. The same base stack runs on all nodes of the ad hoc network. A fully distributed resource location and discovery layer known as Manifold [Dov03] provides a resource location service that can operate in pure ad hoc networks and in fixed networks. Name address resolution is achieved using this layer. The layer uses two mechanisms to deal with queries. A P2P (peer-to-peer) style search with TTL-controlled flooding is used for inexact queries and an overlay network approach (based on the content exposed by the nodes) is used for exact queries. Obviously any ad hoc node must use an ad hoc routing protocol. As mentioned previously in this chapter the protocols that have been implemented for use in this testbed include Dynamic Source Routing (DSR), Optimised Link-State Routing (OLSR) and Ah hoc On-demand Distance Vector routing (AODV). A large number of physical layers have also been designed. The communication stack can sit on a UHF radio, an IrDA link or an 802.11 wireless LAN. Physical layers for interacting with GPS hardware and sensor circuitry also exist. Alternatively, for development purposes, the physical layer can be replaced with a simulation layer that allows multiple nodes to virtually exist on one PC and to define which nodes are in and out of range. In Figure 4.7 the base stack has four physical layers, namely an 802.11 layer, an IrDA layer, a layer for obtaining serial sensor data and a simulation layer called JEmu [Fly02]. The 'up mux' layer facilitates the multiple physical layers. All layers run in the Microsoft Win32 environment. A Windows CE version of everything exists for handheld devices.

The very modular building block like system of DAWN very easily facilitates the experimentation needed to design a multi-protocol network-layer. In particular it facilitates real time reconfiguration as mechanisms can be designed for the DAWN stack that permit the easy real time exchange of the ad hoc routing protocols.

4.8 Summary

The network-layer has always been the core layer of the network; it is the *glue* layer that joins remote hosts and routers together using a variety of different protocols that may emanate from disparate sources and are suited to disparate autonomous systems. As Deering noted, the simplicity of the network-layer's service is its strength; it is relatively easy to replicate the unreliable packet-switching service across the various disparate elements of the internetwork. This attribute of the network-layer has allowed the heterogeneous Internet to develop as it enables more interconnectivity between diverse systems of networks.

However, the history of the Internet has also demonstrated the difficulty of upgrading or modifying the network-layer in a live network when change is required. The protracted migration from IPv4 to IPv6 clearly illustrates the difficulties of changing network-layer protocols when compared to the higher and lower layers. The same network-layer structure has been suggested for use in mobile wireless ad hoc networks in the literature. However, such an approach ignores the fact that the ad hoc network setting is much more challenging; it is a dynamic and diverse environment in which transient ad hoc networks are set-up and torn-down.

Taking account of the experience of legacy networking systems, the outline of a flexible multi-protocol network-layer that addresses the concerns of a mobile ad hoc networking context was presented. This is a layer that addresses the objectives set out in Chapter 3 by means of a suite of components that access a selection of protocols with individual performance strengths and weaknesses. These components rely on an ability to observe networking events, to analyse these events and to execute unilateral and multilateral decisions based on such observations and analyses. A basic blueprint that incorporates these elements of the component design, and which will be used to guide the design of the flexible components throughout the rest of this thesis, was also presented. It is noted that as there are many basic design choices to be made in constructing such a system it is necessary to underpin the design by adhering to a number of basic assumptions and propositions which have been enumerated. The structure of the basic network-layer, which includes the three routing protocols: AODV, DSR and OLSR, some attendant data structures and the experimental Network-Layer packet were also described in this chapter. Finally, DAWN, the platform for the design of this flexible multi-protocol network-layer was described.

5 Enabling Observation & Analysis

5.1 Introduction

As was shown in the survey of ad hoc routing protocol evaluation studies, presented in Chapter 2, the performance of ad hoc routing protocols is dependent on the underlying networking conditions. In particular, it has been shown that key factors influencing the performance of ad hoc routing protocols are the mobility and degree of the nodes and the traffic loading applied to the network.

In the context of flexibility at the network-layer, a flexible node should be capable of observing the dynamics of the network that affect the operation and efficiency of the network-layer so that the node can adapt or reconfigure itself in response to changed conditions. This chapter discusses a node's ability to passively observe the underlying network conditions in a manner that respects the system assumptions and propositions discussed in Chapter 4. The chapter presents three passive, myopic metrics that enable a node to independently measure the node mobility, node degree and traffic loading experienced by a network.

Furthermore, this chapter goes on to describe a set of network scenario trials that have been carried out to ascertain whether nodes can appreciate the different network dynamics of these scenarios using these metrics, and to establish if the performance profiles of the three candidate routing protocols vary under these scenarios, i.e. to identify performance crossover points.

5.2 Observing the Networking Environment

Returning to the blueprint of the flexible multi-protocol network-layer, as illustrated in Figure 5.1, this chapter focuses on the highlighted areas; namely it focuses on the types of network observations that are possible, and the mechanisms for making those observations. The current state of the node, as characterised by its current active routing protocol, influences the observations that are possible.

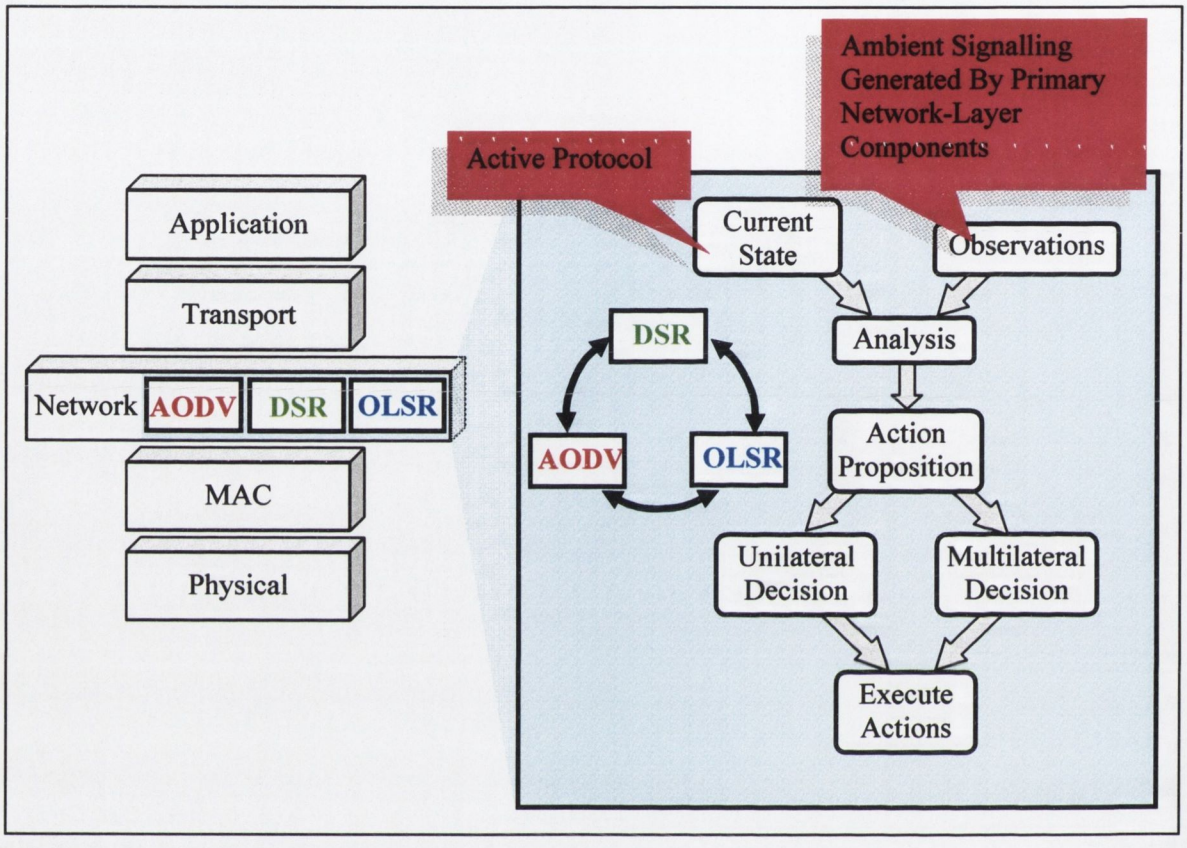


Figure 5-1 Blueprint for Design of Flexibility Components: Network Observation

It has been shown in the extensive and detailed survey of published evaluations in Chapter 2 that the performance of ad hoc routing protocols is primarily influenced by the characteristics of the underlying network. Each of the three candidate protocols has a different performance profile when challenged by varying combinations of the variable network dynamics. The ad hoc routing protocols under review have no control over these network dynamics. They do not have the ability to control the node degree, by varying the transmitter power, or the ability to change the relative speed of the node. The literature has shown that node mobility, node degree and traffic loading are good indicators for the performance of ad hoc routing protocols.

In order to enable the network to adapt itself to changes in the networking conditions that it is experiencing, it is necessary that the constituent nodes of the network should be able to appreciate these networking conditions. As the ad hoc network is an inherently distributed system it should be borne in mind that it is impossible for a node to attain a 'god-like' or global bird's eye view of the entire system. One purpose of this work is to propose a scheme that enables individual nodes to capture information that is correlated to the actual networking conditions and, by inference, to the performance of the ad hoc routing protocols.

While the studies presented in Chapter 2 have shown that the above-mentioned network dynamics influence the operation of the protocols, they do not propose a mechanism that would allow the network's nodes to either capture or quantify these dynamics. For example, the mobility metrics that are proposed in [Joh99], [Kwa03], [Su01] and [Vie03] to enable a mathematical description of the underlying mobility conditions rely on information that is only available because of the artificial nature of the simulated or mathematically modelled networks. The metrics cited in these papers are generally used to capture node mobility information in a single number so that the mobility scenarios may be easily compared, i.e. a single number represents the mobility of the nodes rather than a mobility model such as the RWP model which is defined by multiple parameters.

An example of such a mobility metric is that presented in [Joh99] which attempts to model and measure the mobility of the nodes by looking at individual time-averaged mobility measures. The individual mobility measure between any two nodes is defined as their absolute relative speed taken as an average over time. The total mobility metric for a scenario is the average of the individual measures for each node pair. This metric is made possible as the node velocity information for each node is available to the simulator engine. In reality, the collation of such information in real-time, online, would be dependent on the existence of data aggregation services and external positioning systems such as GPS. However, such a technique would violate a fundamental premise of this thesis: that an ad hoc network, if it is to remain a robust system that is deployable anywhere at any time, must be independent of external networks or systems, and devoid of any reliance on centralised protocols or schemes.

Network management protocols such as the Simple Network Management Protocol (SNMP) [Cas90] and the network management protocols developed specifically for ad hoc networks such as the Ad hoc Network Management Protocol (ANMP) [Che99], the Guerrilla management protocol [She02] and others [Gur98], [Kan01], are used to manage networks at runtime in response to perceived changes in the state of the network. These schemes incorporate network measurement techniques that allow the network to be tuned in response to changes in networking conditions. However, such application-layer schemes are unsuitable for use at, or adaptation to, the flexible multi-protocol network-layer presented in this thesis due their centralised nature and heavy reliance on the multi-hop functionality of the network-layer itself.

Some existing routing schemes, such as those presented in [Lee03b] and [Ram03], use adaptive techniques to tune their operation based on the observation of certain network conditions. The Hotspot Migration Protocol (HMP) [Lee03b] is a scheme that reacts to transient congestion in a network by throttling the rate of TCP flows and the rate of new route requests. Congestion caused by the intersection of multiple TCP streams has been shown to be a problem for AODV and DSR in very active networks. The HMP scheme is heavily reliant on information gleaned from the MAC-layer, such as contention delays and buffer occupancy, to ascertain whether or not the node is indeed a congestion hotspot. HMP acts to reduce congestion by purposefully slowing down the flow of traffic at the hotspot even more. By artificially throttling the traffic throughput rate at the hotspot node even further than the actual networking conditions have, HMP operates on the assumption that the TCP scheme will decrease the flow of traffic from the TCP source in response to the artificial extra congestion at the hotspot node, which the TCP source will perceive as naturally occurring network congestion.

Even though these management and adaptive network-layer schemes are designed specifically for the mobile ad hoc networking environment, they are not in keeping with the objective of developing a robust and flexible system that is as distributed and lightweight as possible. It is therefore proposed that the network-layer observation scheme used to capture information about the networking conditions should fulfil the following requirements which echo the underlying system assumptions and propositions discussed in Chapter 4. Specifically, the network-layer observation scheme should use techniques that are:

- Cross-layer independent
- Distributed and compatible with the network-layer fundamentals
- Good indicators of the performance of ad hoc routing protocols

The network-layer observation scheme should not have any *cross-layer dependencies*; information that is particular to specific MAC layers or transport schemes should not be used to inform the observation mechanism. Cross-layer dependencies place a strait-jacket on the structure of the protocol stack, inhibiting the ability of other layers to be independently adapted, reconfigured or upgraded. For example, the HMP [Lee03b] scheme queries the IEEE 802.11 MAC and TCP to gather protocol-specific data. However, the variety and rate of change of TCP schemes, coupled with the development of alternative non-TCP-based transport schemes dedicated to the mobile wireless ad hoc networking environment, would possibly render such a scheme inoperable if the dependent components were upgraded.

Nodes are also likely to face more short-term changes when other layers in the stack are dynamically reconfigurable. Consider an ad hoc node with multiple physical-layer options, i.e. one that is equipped with both an IEEE 802.11 radio and a Bluetooth radio. (Incidentally, the existence of multiple interfaces is specifically supported in AODV, DSR and OLSR.) When the node is in an open environment such as a street or a park it may be preferable to use the IEEE 802.11 radio as it has a longer range, enabling connectivity between relatively distant nodes. If the node was moved indoors, to a café or an office, it may be preferable to use the shorter range Bluetooth radios as they enable more nodes to operate within a smaller area. For such a change to be possible the node should be able to hand-off between the two radios without regard for the needs of higher layers; transport schemes ensure that lost data packets are retransmitted and in many cases, such as for streaming video and audio, some loss is tolerable at the application layer. If the operation of the higher layers was dependent on the specific architecture of the underlying radios, such as the queue lengths or the MTU, rather than the basic one-hop connectivity service they provide, then the flexibility of the higher layer would be compromised.

The scheme should also be *distributed and compatible with the network-layer fundamentals*, i.e. it should not compromise the inherent qualities of the ad hoc networking protocol in use by introducing signalling that is incompatible with the operation of the underlying routing protocol. Ad hoc routing protocols are inherently distributed in their operation; centralised system models are unsuitable as the mobility and link characteristics of mobile ad hoc networks consistently undermine hierarchical structures. As such, a scheme that operates at the network-layer should be as robust as the routing protocols that it supports.

In broadcast networks, both the receipt and transmission of packets have an energy and time cost, the minimisation of which is key to designing any system for wireless ad hoc networks. The ad hoc routing protocols have been designed to minimise control overhead, specifically the number and size of transmitted control packets, in a trade-off with efficiency of operation. Protocols such as DSR and AODV are reactive by design; they throttle back the amount of overhead incurred when demand for routing services is low. Protocols such as OLSR, on the other hand, use periodic control message transmissions to gain up-to-date network topology knowledge regardless of the demand for routing services.

As the aim of this system is to leverage the existing optimisations of each ad hoc routing protocol by designing a multi-protocol system which should allow the routing protocols to be changed at run-time, the signalling characteristics of the system will change. Accordingly, the system should be designed in such a manner that it adapts to the changes in the primary network-layer components. For the design of the flexible system to be compatible with the fundamental properties of the underlying network-layer components, the scheme should be passive, i.e. each node should not add to the signalling profile of the network-layer. Rather, the scheme should use the existing network-layer activity to inform its observations regarding node mobility, node density and traffic loading.

Finally, the scheme should use metrics that are *good indicators* of the performance of the ad hoc routing protocol. In order for a metric to be a good indicator of protocol performance, the metric should capture and reflect the pertinent characteristics of the networking conditions that affect the performance of the ad hoc routing protocol. For example, it has been noted that node mobility affects the performance of routing protocols and while much of the literature characterises node mobility in terms of absolute velocities and pause times, it has been shown that it is the relative mobility of nodes, and the associated link stability, that affect the performance of the protocols. Therefore, from the point-of-view of measuring node mobility, the relative mobility of a node is a good indicator of the performance of the protocol rather than the absolute mobility of a node.

5.2.1 Observable Events

Given the criteria outlined above, and the earlier assumptions and propositions outlined in Chapter 4, it is possible to identify a number of sources of information that meet the system's requirements. The scheme aims to use low-cost techniques that are based on each node leveraging information from events that it may observe in the course of operating an ad hoc routing protocol, i.e. the receipt, issuing and forwarding of data and control packets and the occurrence of *notifiable network-layer events* within the network-layer itself. It is assumed that the network-layer is able to eavesdrop on all packets that are presented to the radio interface of the node. Such an assumption is not unusual in the development of wireless ad hoc networking systems. When a device is operating in so-called promiscuous mode every packet is received at network-layer without being filtered based on link-layer addresses or other identifiers. This approach has already been taken by the designers of DSR and AODV which make use of all available information that they may glean from packets received from the lower layers, regardless of whether or not the packet is intended for the node or not.

While it has been proposed that the network-layer observation scheme should not be reliant on cross-layer feedback, it is also important to limit the amount to which the scheme is reliant on feedback from any

particular element of the network-layer itself. The interaction between the observation scheme and the network-layer should be as abstracted as possible, i.e. the events that feed back into the scheme should be common to all elements of a network-layer implementation. This approach is necessary to allow for the reconfiguration of components at the network-layer, i.e. where one component may be replaced by another more suitable component of similar functionality. For example, every node maintains a route cache which adds or removes route entries according to the ad hoc routing protocol's rules, while nodes also maintain protocol-unique state tables which maintain information that is only pertinent to the given protocol, and which may vary from node to node.

A notifiable network-layer event is an event that occurs naturally at the network-layer, i.e. within the current routing protocol or within the route cache or some other network-layer component. There are a number of events that are common to all network-layer implementations, regardless of the routing protocols or route cache structures in use. Among these common events are the discovery and loss of neighbouring nodes and the request of higher layers for data forwarding services. When such events take place, it is of negligible cost to the network-layer to set a flag within the layer or to send a signal from one component of the layer to another. Furthermore, the modification of routing protocols, data forwarding mechanisms or route cache structures to issue such notifications does not alter the operation of the ad hoc routing protocols or network-layer in any way.

It may be noted at this stage that the imposed conditions dictate that the node only has a *myopic* view of the surrounding environment, i.e. it can only observe events local to itself or events that cause neighbouring nodes to issue packets. This is in contrast to the bird's eye network view of a simulation engine, most often cited in the published literature. Among the events that an individual node can observe without breaking any of the assumed restrictions outlined above are:

- *Common network-layer events:* The times at which neighbouring nodes are discovered and lost.

  Whenever a packet is passed to the network-layer from lower layers, the packet header contains information identifying the immediate last-hop source of the packet transmission. The network-layer notifies a **NEIGHBOUR DETECTED** event, specifying the neighbour's address. Whenever a network-layer component such as the routing protocol marks a neighbour as lost or inactive it notifies a **NEIGHBOUR LOST** event, specifying the neighbour's address.

- *Traffic events:* The traffic streams that a node is currently trying to originate or support as a source or intermediary node.

  Whenever a packet is presented at the network-layer interface from a higher layer so that it may be forwarded to a specific destination, such an event can be notified as a **DESTINATION SOUGHT** event. The destination address is specified in a traffic event notification. Whenever a node, as an intermediary node or target node, receives a data packet from a neighbouring node that it should forward to another node or pass to a higher layer, it notifies a **DESTINATION SUPPORTED** event. Again, the destination address is specified in a traffic event notification. It does not matter if the node forwards data packets in

accordance with the specifications for IPv4 routers [Bak95] as AODV and OLSR do, or if it uses another mechanism such as DSR's source routing.

5.2.2 Passive, Myopic Observation

The following paragraphs outline the metrics that have been chosen to capture the node mobility, node degree and traffic loading characteristics of ad hoc networks using either AODV or DSR or OLSR. These metrics conform to the criteria laid out above. It should be noted that the use of such independent, passive metrics to measure a network's dynamics is novel.

■ **Node Mobility Metrics**

Bearing the suggested requirements in mind and the data that is available to an individual node, certain mobility metric candidates can be ruled out. The artificial parameters of the RWP model such as node velocity and pause time cannot be used as these parameters are not normally available in real networks without recourse to systems that can provide velocity information. Furthermore, these parameters are not, in themselves, good indicators of the performance of a protocol as they do not capture the aspects of node mobility that have been found to affect the operation of ad hoc routing protocols. The requirement for the metric to be computed in a real world implementation rules out the use of a metric such as the mobility metric described in [Joh99]. That mobility metric was dependent on nodes knowing the speed of their neighbours so that they could compute their average mobility relative to their neighbours'. The availability of such information in a real system is dependent on the use of positioning systems such as the GPS and out-of-band signalling or data aggregation services to relay the information to a node.

As noted in Section 2.4.5, it is the relative mobility of nodes that has a direct bearing on the stability of links between neighbouring nodes. The stability of the links influences the rate at which they come up or go down, i.e. the rate of one-hop link changes. It has already been shown in both [Joh99] and [Vie03] that there is an approximately directly proportional relationship between the rate of link changes and the increasing relative mobility of the nodes. Boleng *at al.* [Bol02] have examined criteria for mobility metrics that may enable the development of adaptive ad hoc routing protocols. The authors suggest the use of average link duration as a metric to capture relative mobility information about ad hoc nodes. However, the link duration information is not captured from the network per se; rather, it is taken from information available to a simulator engine which indicates when two nodes are within transmission range of each other. In reality, nodes will not be able to gather such information about the theoretical duration of the links to each of their possible neighbours unless they are engaged in finely tuned link sensing. Using the information available to nodes employing the proposed network-layer observation scheme, it is possible to observe both the average link-change rate and the average link-duration by recording the `NEIGHBOUR DETECTED` and `NEIGHBOUR LOST` events which are notified by other network-layer components.

■ **Node Degree Metric**

The observation of a node's degree is relatively straightforward. In networking terms, the collection of nodes that are within one transmission hop of a node are its neighbours. Using the information available to nodes

employing the proposed network-layer observation scheme, it is possible to observe the average node degree by recording the `NEIGHBOUR DETECTED` and `NEIGHBOUR LOST` events which are notified by other network-layer components.

■ Traffic Loading

Using the information available to nodes using the proposed network-layer observation scheme, it is possible to observe the average destination-diversity of the traffic loaded onto the network by recording the `DESTINATION SOUGHT` or `DESTINATION SUPPORTED` events at each node. The ad hoc node is challenged by the number of the destinations to which it is expected to provide routes for traffic that is loaded onto the network and the length of time that it is required to route traffic to those destinations. As discussed in the summary of published evaluations in Section 2.4.5, reactive protocols are primarily taxed by the number of traffic streams that they are expected to support. Reactive routing protocols must engage separate route discovery processes for each unique destination unless the information is already cached. Furthermore, intermediate nodes must support established routes by engaging the route maintenance mechanisms. The negative effects of increased node mobility on the performance of the reactive routing protocols are exacerbated by the increased diversity of the destinations that are being sought or supported at anytime. On the other hand, OLSR has been designed to support requests for multiple destinations as no additional overhead is incurred. Therefore, the destination-diversity metric records the average number of distinct destinations that a node is seeking a route to or supporting a route to.

■ Summary

With regard to both the observation of node mobility and node degree, it should be noted that the proposed scheme is dependent on the ad hoc routing protocols in use. The observations that can be made about node mobility and node degree are dependent on the frequency of the notifiable network-layer events, which in turn are related to the frequency of the packet transmission activity in the node's neighbourhood. As the activity of nodes at the network-layer is dependent on the routing mechanisms employed, the amount of data packets that appear on the ether depends on the ad hoc routing protocol in use.

The three protocols have varying signalling characteristics which affect the amount of data that is observable at any given time. In a purely reactive protocol such as DSR, the level of activity on the ether is directly related to the traffic that is loaded onto the network. If a node is required to discover a path, either for itself or for another node, then the node will generate control traffic. Nodes along an active path in a DSR-enabled network will continue to be active as long as the communicating source and destination nodes continue to forward data packets. If the network becomes less active as nodes terminate their end-to-end communications, then the level of activity on the ether will die off. The fall-off in the amount of data or control packets on the ether should not negatively affect the operation of the scheme. It should be noted that a silent or inactive network using a reactive routing protocol is an optimal proposition as it uses the least resources. The lack of observable data on the ether is informative in itself.

AODV uses a mix of reactive and proactive elements in its route discovery and route maintenance mechanisms. It employs a proactive beaconing mechanism along active paths that means that active nodes

will issue periodic packets which may be observed by neighbouring nodes. As with DSR, inactive nodes in an AODV-enabled network do not generate any traffic at the networking layer, so that inactive regions of AODV nodes experience the same lack of observable data that inactive DSR nodes do.

On the other hand, OLSR actively maintains a neighbourhood table at each node in compliance with the protocol specifications. As OLSR is a link state protocol it frequently monitors the state of it neighbourhood links, OLSR nodes are provided with a constant stream of network-layer events which they may observe regardless of the traffic activity on the network

5.3 Indicative Trials

A series of trials were conducted with two main goals in mind. The first goal of the trials was to try and exact a better understanding of the performance profiles of the three candidate protocols under a selective variety of stressing conditions. The second goal of the trials was to assess whether the individual nodes could observe enough indicative information using the passive techniques to infer pertinent information about the state of the network under those conditions. The trials were set up using the JEmu [Fly02] simulator. JEmu uses a simple radio model which allows for the degradation of the transmitted signal strength according to the inverse square law. In practical terms, the signal strength of a transmitted packet is quantised into two levels which translate into two transmission distances. This means that each transmitted packet has a transmission range and an interference range. The transmission range is the distance that a packet may travel and still be above the required SNR level for the receiver. The interference range is the distance that a packet may travel and still have enough power to interfere with another signal. If two packets interfere with each other, i.e. if two packets are transmitted at the same JEmu quantisation time within interference range of each other, then the interfering packets are dropped by JEmu. The trials were all conducted using a population of 50 nodes. The transmission coverage area of each node was 31416 m^2, based on a transmission range of 100m. The simulation space, node mobility and traffic model parameters were varied to generate a variety of testing scenarios. Each individual simulation was run for 900s.

■ Node Mobility Model

A random waypoint (RWP) mobility model was used in the standard trials. The RWP mobility model, as described in Chapter 2, is tuned using the following parameters: minimum velocity, maximum velocity and pause time. Three variations of the mobility model were evaluated: a low mobility model, a moderate mobility model and a high mobility model. The values used to model these scenarios are given in Table 5.1. As may be seen in Table 5.1, the pause time for each scenario was varied from 0s (i.e. constant motion) up to 120s. The RWP model was used in these trials as it is places equal stress on all of the node-to-node links. While the RWP model may not capture or recreate a realistic scenario, it serves its purpose as every real scenario consists of a series of node-to-node links which have the same stability properties as some variant of the RWP model. The stability of the node-to-node links is the primary way in which the relative mobility of nodes affects a node's performance by demanding rapid route discovery and maintenance techniques from the routing protocol.

Three mobility models have been chosen to determine whether nodes can independently distinguish between scenarios. The low mobility model was chosen to represent a relatively stationary environment experiencing some light mobility which may be characterised by a mix strolling pedestrians and stationary objects. This mobility model was used in [Bro98], [Das98] and [Joh99]. A moderate mobility model was chosen to simulate a more challenging range of motion corresponding to brisk walkers, cyclists and slow urban vehicular traffic. Similar models were used in [Cla02b], [Das98] and [Per02]. The high mobility model is a much more demanding scenario which corresponds to cyclists and vehicular traffic. Similar mobility models were used in [Bro98], [Das01], [Joh99] and [Per02].

Table 5-1 RWP Mobility Model Parameter Values

Mobility Model Parameter Values	Minimum speed (m/s)	Maximum speed (m/s)	Pause Time (s)
Low mobility	0	1	0, 30, 120
Moderate mobility	1	5	0, 30, 120
High mobility	5	20	0, 30, 120

■ **Simulation Area (Node Degree) Model**

The simulation space was a simple square area. Three different dimension settings were used: $350m^2$, $500m^2$ and $750m^2$. The corresponding node density and node degree values are shown in Table 5.2. As explained in Chapter 2, node degree is the measure of how many neighbours a node has.

Table 5-2 Simulation Area, Node Density and Node Degree Parameter Values

Simulation Area Parameters	Simulation Area (m^2)	Average Node Density (Node/m^2)	Average Node Degree (Neighbours) ***[Rounded Up]***
Low density	$(750)^2 = 562,500$	1 node / $106m^2$	2.79 [*3*]
Moderate density	$(500)^2 = 250,000$	1 node / $70.7m^2$	6.28 [*7*]
High density	$(350)^2 = 122,500$	1 node / $49.5m^2$	12.8 [*13*]

The values used for this setting are chosen to reflect three quite different spaces. The low density model creates a sparsely populated network in which the average node degree is about 3 neighbours. A similar low density model was used in [Bro98]. The moderate density model yields an average node degree of 7 neighbours. Similar density models were used in [Cla02b] and [Joh99]. The high density model yields an average node degree of 13 neighbours. Similar density models were used in [Das98], [Das01] and [Per02].

■ **Traffic Model**

The traffic models used in the experiments were based on a combination of constant bit rate (CBR) sources and sinks. By varying the number of source-sink pairs, or streams, and the rate at which the pairings change, the efficiency of the routing protocols' mechanisms was challenged. Two traffic scenarios were modelled, as detailed in Table 5.3. All traffic sources originate CBR traffic, packets of 64 bytes at a rate of 4 packets/s.

Table 5-3 Traffic Model Parameter Values

Traffic Model Parameters	Traffic Model Description
Sporadic	Each node randomly chooses a destination node, using a uniform distribution, from all of the other nodes and communicates with that node for a random period of time before choosing a new destination node
Recurrent	20 pairs of source-destination nodes are chosen. There is continuous traffic between each pair. The pairs do not change for the duration of the simulation.

■ **Collation of Data**

As each simulation was run, each node logged events to a local repository. Nodes time-stamped events which relate directly to the myopic metrics described earlier. They also logged information regarding the control and data packets that were issued, forwarded and received by nodes, allowing for the calculation of global performance metrics. Those logged events are analysed from both the myopic and global viewpoint. Both sets of metrics are evaluated over the entire duration of the simulation, yielding a timed series of data. Thus, for any instant during the simulation, a record of the actual global network performance of the node and its instantaneous local view are available. In reality, the record of the actual global network performance is only ever known to the simulator engine whereas the myopic view is always available to each node.

5.4 Performance Profile Comparison

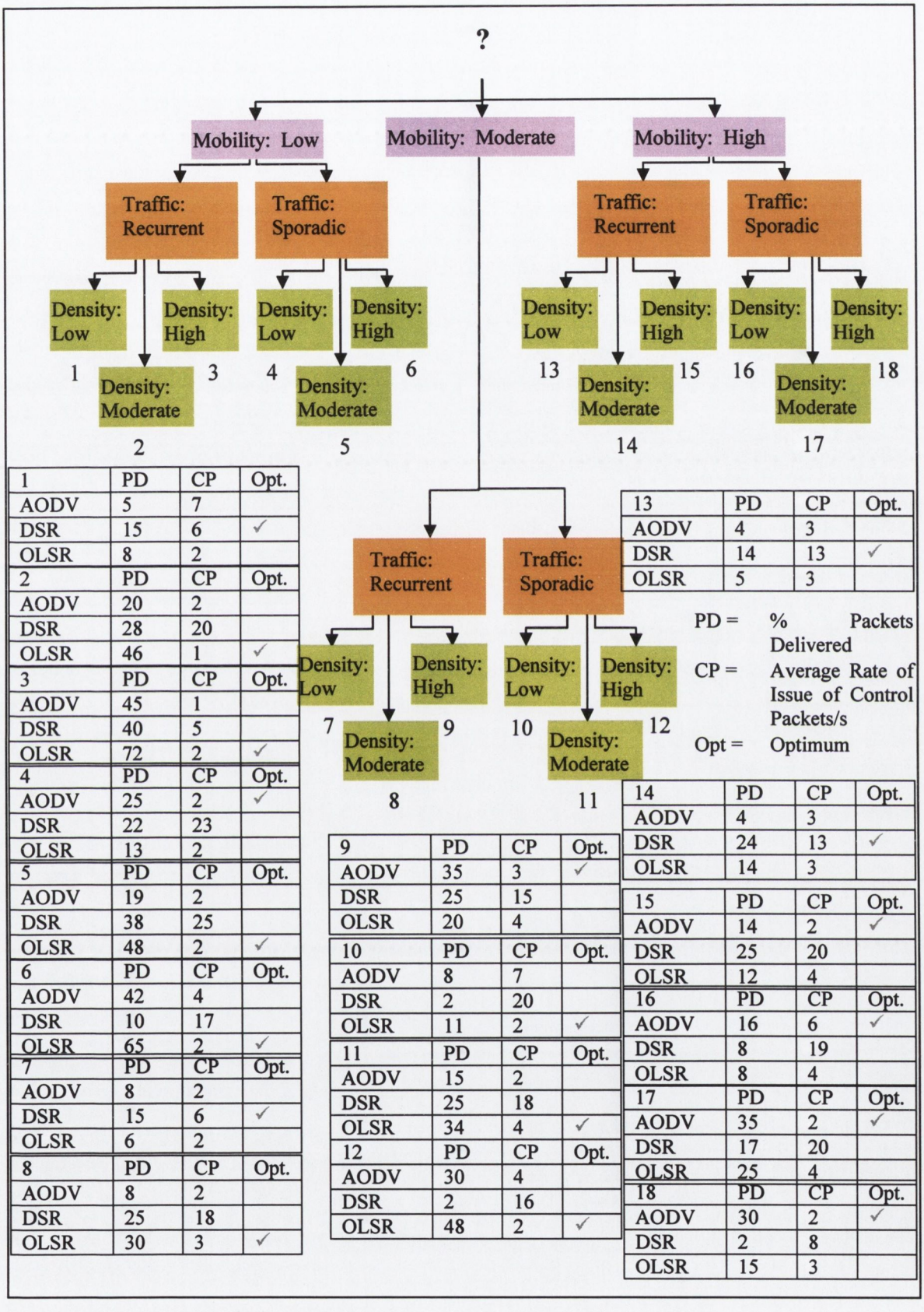
The purpose of evaluating the performance profiles of AODV, DSR and OLSR networks under the conditions outlined in Section 5.3 was to ascertain if there is an appreciable difference between the profiles of the three ad hoc routing protocols under those conditions and in doing so, to identify which of the ad hoc routing protocols is optimum under the specified trial conditions.

The objective of each routing protocol is to maximise the network throughput while minimising the control overhead incurred. Ad hoc routing protocols also seek to minimise the end-to-end packet delay. Delay is incurred in two situations. The first scenario in which a data packet may be delayed is when it is first presented to the network-layer by a higher layer. At this point, the network-layer's route cache is queried for a route to the specified target node. If a route is present, the data packet is packet-switched immediately. However, in AODV and DSR networks, if a route is not already cached, then the data packet is stored in a FIFO Send Buffer pending the reactive discovery of a route. This leads to an inevitable delay due to the route discovery latency period. If a route is eventually discovered, which neither AODV nor DSR can guarantee to

be the shortest route available, then the packet is switched across the network using the established route. Furthermore, both protocols check that each packet is forwarded to the next-hop node using a variety of acknowledgement techniques. Pending the receipt of some type of acknowledgement, both AODV and DSR nodes store the packet in a FIFO Maintenance Buffer while they try to forward the packet again. This approach may yield an increase in the end-to-end delay experienced by the data packet.

On the other hand, an OLSR node which receives a data packet for which it does not have a route will discard the data packet immediately. If the OLSR node does have a route, it will be the shortest route possible as OLSR uses Dijkstra's shortest-path algorithm to calculate routes. Dropped packets do not yield an end-to-end delay record, which favours the OLSR end-to-end delay performance profile. Furthermore, if the route breaks down at an intermediate node, OLSR discards the data packet immediately. Again, such a packet does not yield an end-to-end delay record as it is not delivered. The literature has shown that the results for OLSR always outperform those protocols that buffer data. Therefore, end-to-end delay results are not used in the comparison of the protocols' performances. The results presented in Figure 5.2 represent the averaged results for each of the mobility models.

DSR may be seen as the weakest protocol under sporadic traffic loading for all density and mobility scenarios. DSR's strength lies in low density environments with recurrent traffic, a scenario in which the other protocols fail. OLSR is the weakest protocol in the high mobility scenarios, regardless of any other scenario characteristics; OLSR cannot converge quickly enough in a high mobility environment. On the other hand, when faced with sporadic traffic and moderate to low mobility scenario, OLSR is the most efficient protocol, except for low density environments in which case AODV is the better performer. Indeed, for all low density scenarios, OLSR has a very poor performance profile. The lower node mobility affects OLSR's ability to disseminate topology update messages quickly enough. In a low density environment, OLSR loses its optimising feature, i.e. the selection of multi-point relay nodes, and it returns to being a simple link-state protocol. AODV is also affected by lower node densities due to its highly distributed distance-vector technique which does not allow the caching of extra routing information. On the other hand, the light control overhead incurred by AODV makes it the best performer in almost all of the high density environments. OLSR is generally the most efficient protocol in high density networks at low to moderate speeds.



1	PD	CP	Opt.
AODV	5	7	
DSR	15	6	✓
OLSR	8	2	

2	PD	CP	Opt.
AODV	20	2	
DSR	28	20	
OLSR	46	1	✓

3	PD	CP	Opt.
AODV	45	1	
DSR	40	5	
OLSR	72	2	✓

4	PD	CP	Opt.
AODV	25	2	✓
DSR	22	23	
OLSR	13	2	

5	PD	CP	Opt.
AODV	19	2	
DSR	38	25	
OLSR	48	2	✓

6	PD	CP	Opt.
AODV	42	4	
DSR	10	17	
OLSR	65	2	✓

7	PD	CP	Opt.
AODV	8	2	
DSR	15	6	✓
OLSR	6	2	

8	PD	CP	Opt.
AODV	8	2	
DSR	25	18	
OLSR	30	3	✓

9	PD	CP	Opt.
AODV	35	3	✓
DSR	25	15	
OLSR	20	4	

10	PD	CP	Opt.
AODV	8	7	
DSR	2	20	
OLSR	11	2	✓

11	PD	CP	Opt.
AODV	15	2	
DSR	25	18	
OLSR	34	4	✓

12	PD	CP	Opt.
AODV	30	4	
DSR	2	16	
OLSR	48	2	✓

13	PD	CP	Opt.
AODV	4	3	
DSR	14	13	✓
OLSR	5	3	

14	PD	CP	Opt.
AODV	4	3	
DSR	24	13	✓
OLSR	14	3	

15	PD	CP	Opt.
AODV	14	2	✓
DSR	25	20	
OLSR	12	4	

16	PD	CP	Opt.
AODV	16	6	✓
DSR	8	19	
OLSR	8	4	

17	PD	CP	Opt.
AODV	35	2	✓
DSR	17	20	
OLSR	25	4	

18	PD	CP	Opt.
AODV	30	2	✓
DSR	2	8	
OLSR	15	3	

Figure 5.2 Sample of Global Performance Results for AODV, DSR and OLSR

5.5 Myopic Measurements

The measurements of node mobility and node degree are dependent on a variety of node and network attributes which have already been noted. Specifically, the measurements are dependent on the routing protocol and, for reactive protocols, whether or not the node is actively seeking or supporting a route to a destination. The node mobility and node degree observations are discussed in tandem as it has been found that a node's observation of its node degree is influenced by its relative mobility to other nodes. An AODV node does not proactively beacon signals to its neighbourhood unless it is part of an active route, i.e. it is supporting a route to a specific destination. In that case, an AODV node issues periodic beacons to indicate that it is still there. This forms part of AODV's proactive route maintenance technique. Otherwise, nodes only detect the existence and the failure of links as a result of the signalling generated during the route discovery, route maintenance and packet-switching mechanisms which are demand-driven. In light of this fact, AODV nodes are classified as either active or inactive with regard to their signalling profile. An active AODV node is one that is issuing signals, i.e. control or data packets, as a result of demands that have been placed on the network. An inactive node is one that does not issue such signals. DSR nodes do not engage in any proactive signalling at any time. DSR nodes only detect the existence and the failure of links as a result of the signalling generated during the route discovery, route maintenance and packet-switching mechanisms which are demand-driven. Like AODV nodes, DSR nodes are classified as either active or inactive with regard to their signalling profile. In OLSR, the detection of link-states is a primary component of the routing protocol's technique to sense the neighbourhood and disseminate link-state information to all other nodes so that they may construct network topology tables to populate their route caches. OLSR nodes may be considered to be permanently active with regard to their signalling profile.

The results presented in Figure 5.3 – 5.5 are based on the toughest mobility model, i.e. the 0s pause time. This model places the most stress on the node-to-node links. While the y-axis is self-explanatory for each experiment, the number along the x-axis for each graph corresponds to individual rounds of tests, the averages of which are shown on the graph. So, for each combination of mobility model and density model, nine rounds of tests were carried out. As may be expected, OLSR nodes exhibit the most fidelity to the actual simulation parameters. A common theme across the results is that each of the protocols reports an increase in their node degree when the node mobility increases. The reason for reporting a higher node degree than the simulated average is that the nodes take too long to remove nodes from their caches in high mobility environments. It may be seen that DSR is the worst offender as regards maintaining nodes in its cache. However, it is also evident that active DSR nodes, i.e. those that are engaged in routing traffic have a better estimation of the node degree as their attempted use of neighbouring nodes act as a form of link-sensing and failed nodes are removed more quickly. AODV, which has a harsher route cache timeout policy than DSR, but which does not engage in the active link-sensing of OLSR, delivers almost as much accuracy as the more costly OLSR.

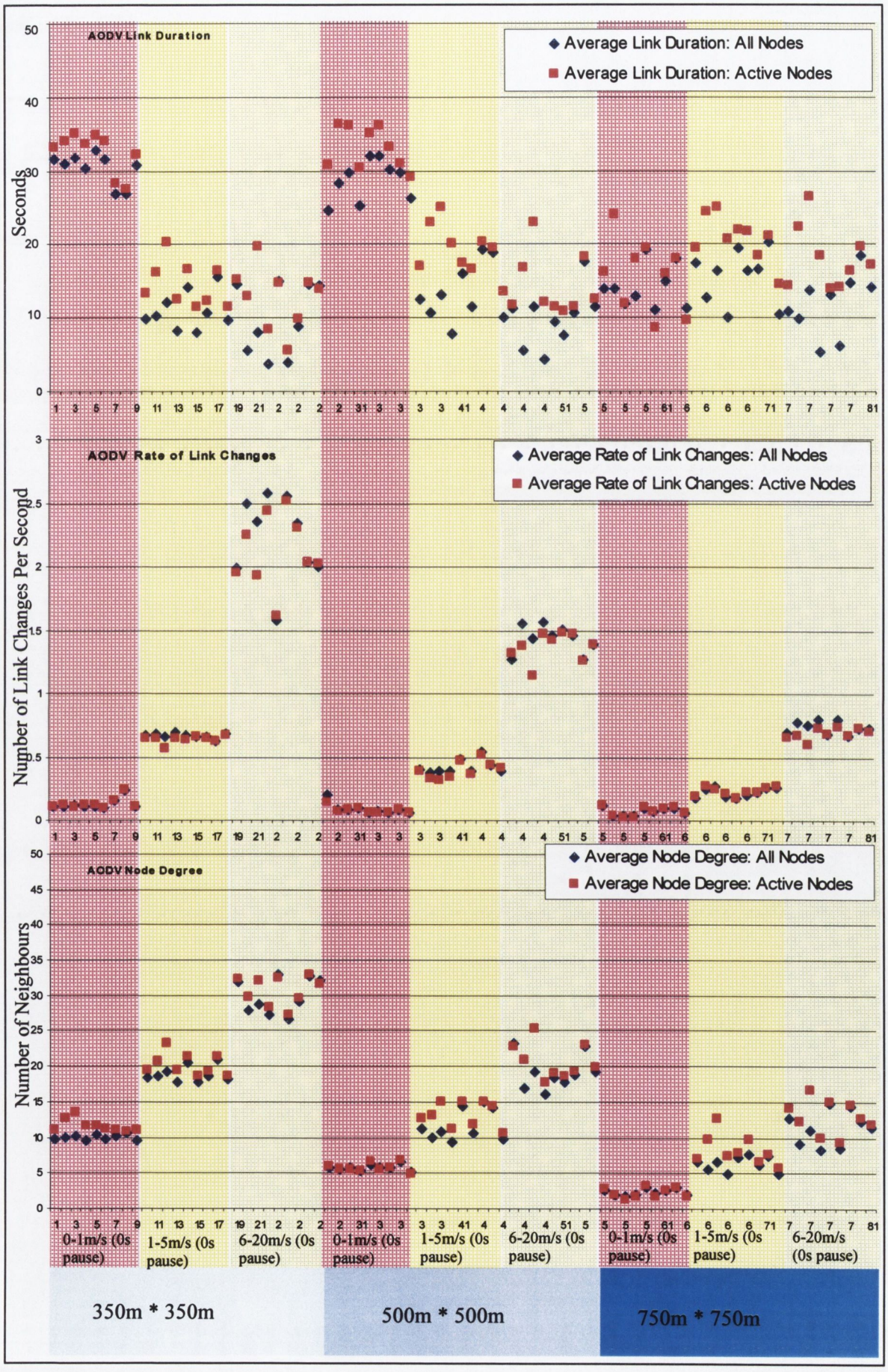


Figure 5.3 AODV Node Degree, Link Changes and Link Duration

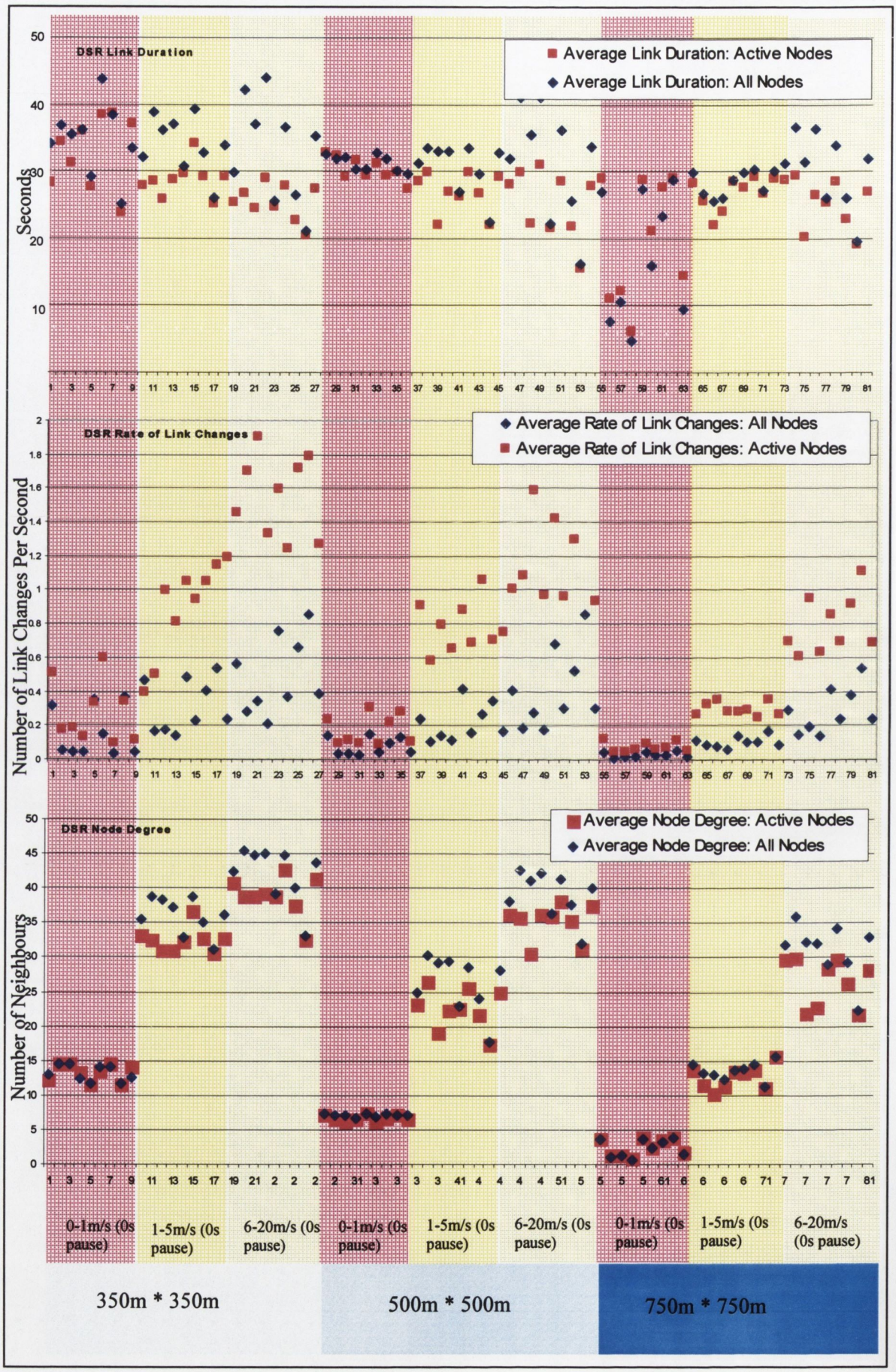


Figure 5.4 DSR Node Degree, Link Changes and Link Duration

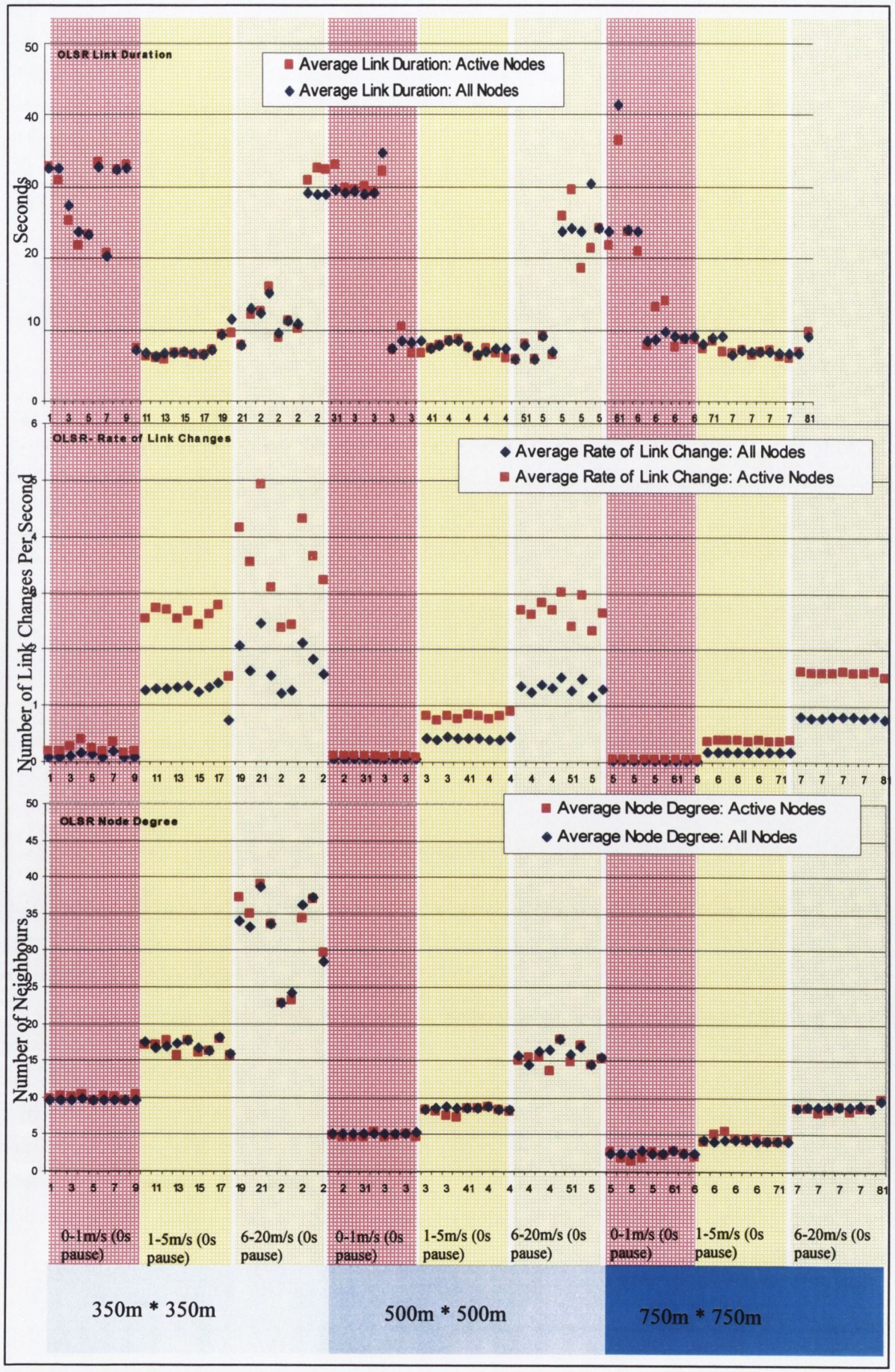


Figure 5.5 OLSR Node Degree, Link Changes and Link Duration

As regards average link duration as a metric for measuring the relative mobility of nodes, it may be seen that there is no correlation between average link duration and node mobility for DSR. This is as a direct consequence of the extended lifetime that DSR affords cached routes. Only OLSR, and to a lesser extent AODV, show a correlation between increasing node mobility and decreasing average link duration. However, it may be seen for OLSR that it cannot distinguish between a moderate and high mobility environment, under any density model, using average link duration. As OLSR is particularly prone to the effects of increased node mobility, such a metric would not suffice. However, turning to the average rate of link change metrics, it may be seen that OLSR can distinguish between different mobility scenarios based on its observed rate of link changes. The same is true for both DSR and AODV which each demonstrate an ability to distinguish between the different mobility models. However, in the case of DSR there is a difference between what an active node measures, i.e. one issuing and possibly receiving more signals, and the average DSR node.

It may be observed that for some of the evaluations, there are relatively large variations in reported averages. For example, the DSR average link duration for active nodes in the high density, low mobility setting varied from 24s to 39s. It has an arithmetic mean of 32.5 and a large standard deviation of 5.2. In order to establish if the cause of these large variations was caused by the choice of runtime (900s), a number of further evaluations were carried out to establish whether the use of longer runtimes would result in a smaller standard deviation.

The reason for this approach stems for the observation that the greatest variance occur in scenarios in which there are fewer events to sample than in others. To clarify, consider the difference between the reported average link duration figures for DSR and OLSR. In the case of DSR, the link duration averages show a large variance, regardless of the experiment settings, indicating that a possible cause of the larger variance is due to the lack of notifiable link change events (the DSR timeouts mean that it does not remove old neighbours as soon as the other protocols). When the infrequency of these events is viewed in conjunction with the runtime length, it is apparent the lack of such events may cause such large variations.

A sample set of extended evaluations (1200s and 1800s) was carried out for each protocol. For each protocol, one set of the conditions was repeated, the high density, high mobility experiment. These experiment settings were chosen as the 900s runtime evaluations of the protocols under these scenario settings yielded large variations in the reported results. Referring to Figure 5.6, it may be observed that the longer runtimes do result in more equilibrium in the results. The longer runtimes, particularly the 1800s, yield a much smaller standard deviation for the both the DSR link duration and link changes. In the case of DSR average link duration, the arithmetic mean is and the standard deviation is., much tighter than the 900s results. However, it should also be observed that while the longer runtimes do ensure that better averages are yielded, they do not enable the nodes to more accurately reflect the real noed degree and node mobility values. These variations between the real node degree and perceived node degree are not dependent on the runtime length, but rather on factors that were outlined earlier.

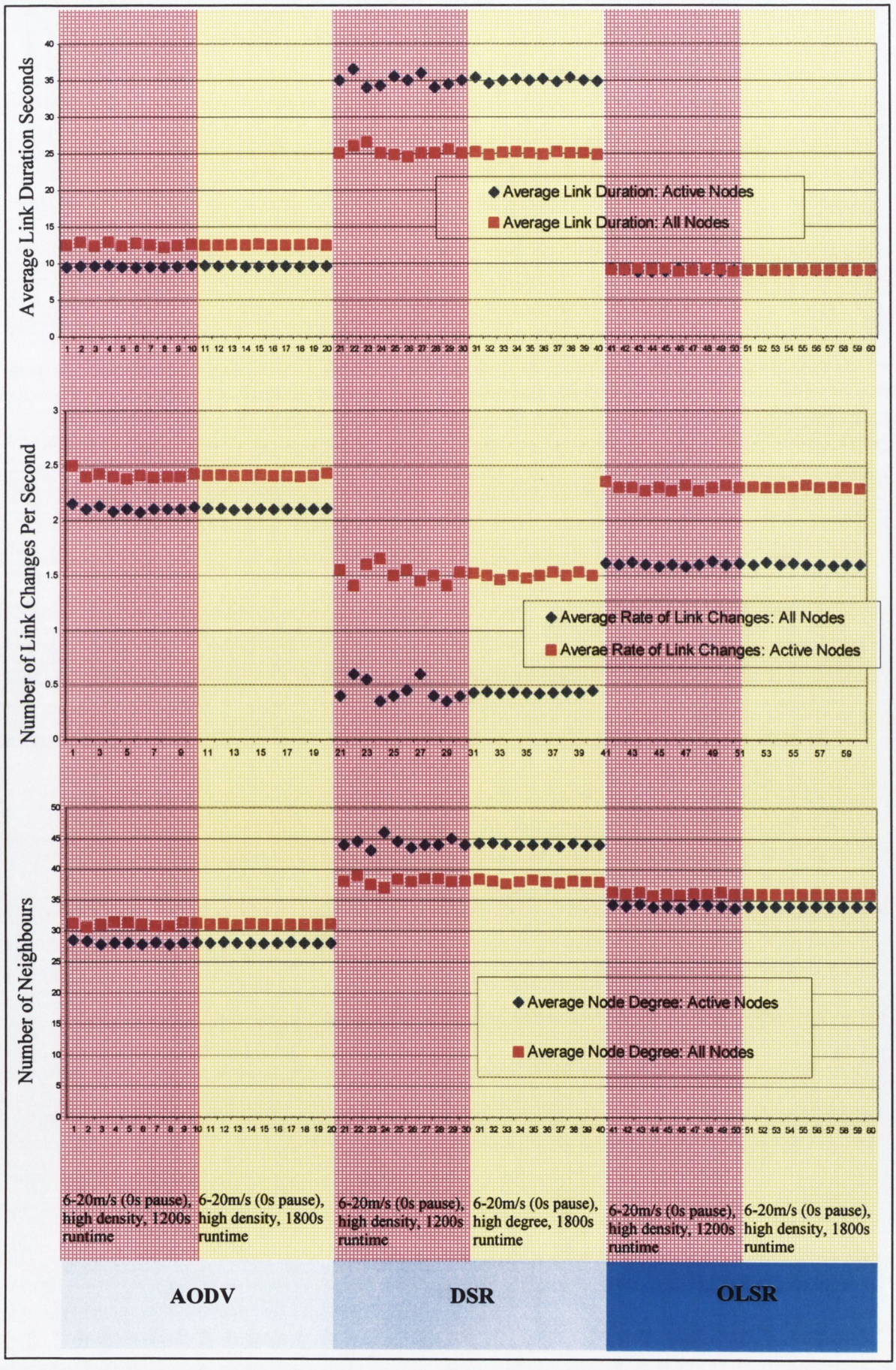


Figure 5.6 Extended Evaluations: 1200s and 18000s

In all, the metrics demonstrate that there is a lot of information available to the nodes. While the performance results presented above have been calculated using offline techniques, each node can perform the same myopic measurements itself at run-time. Nodes store the observed data for node mobility, node density and traffic loading in an External State Vector (ESV). As the networking conditions are subject to change as the nodes move about in an autonomous manner and the level of traffic loading changes, nodes should be enabled to distinguish between short-term and long-term changes as the network will only be reconfigured for lasting changes.

In sum, the ability to distinguish between transient changes and more diuturnal ones using a filter is important. As a node does not have access to any baseline data for the environment that it is operating in, it must use adaptive filters suited to this environment. [Kim01] has investigated the suitability of a number of filters for network measurement tasks. The authors have argued that an exponentially-weighted moving average Flip-Flop filter controlled by an x-chart as proposed in [Rig94] is the most suitable filter for the mobile ad hoc networking environment. The Flip Flop filter provides the combination of both agility and stability. The filter's weighting parameters can be tuned to suit the fidelity of the underlying routing protocol's measurements.

5.6 Summary

This chapter has developed the concept of enabling a node to observe the environment in which it is operating. It was shown in Chapter 2 that the ad hoc routing protocols are primarily influenced by networking conditions that are completely outside their control. Specifically, it has been shown in the literature that nodes are influenced by the relative mobility of nodes, the degree of the nodes and the traffic that is loaded onto the system.

In Chapter 3 it was proposed that nodes should be enabled to observe those conditions so that they could be fed back into an adaptive mechanism that would allow the network to change its routing protocol. However, there are strict conditions underlying the design of flexibility enabling mechanisms in this work and none of the conventional network measurement techniques could be used. Rather, it was determined that the node should use a completely distributed technique that would not rely on active probing of the network. Nor would it depend on the use of the multi-hop routing capability of the network.

In light of this a number of metrics were identified to measure the node mobility, node degree and traffic loading on the system. These metrics make use of the information which is available to the node by means of ambient signalling. The proposition and evaluation of such metrics, which are based on completely distributed and local information is novel. In order to evaluate the use of these metrics and to further understand the performance of the three candidate routing protocols, a series of trials were carried on the DAWN testbed. The trials indicate that the proposed metrics may be used by a node to observe and appreciate its networking conditions. However, in light of the fidelity of some of the measurements, the use of adaptive filters is recommended. Furthermore, a performance comparison of the three protocols identified a number of distinct scenarios in which one protocol is much more suited to the environment than any other protocol.

6 Enabling Auto-Configuration

6.1 Introduction

As outlined in Chapter 3, a key requirement of flexibility in a multimodal ad hoc network is that of auto-configuration of the network-layer. This chapter introduces and explores the issues involved in enabling flexible nodes to freely roam throughout a heterogeneously configured network. As the nodal composition of the flexible ad hoc network, from the point-of-view of node configurations, may change over space and time, nodes that move from region to region or nodes that are booted up after a period of inactivity must adapt to the prevailing regional configuration.

The following sections of this chapter will detail the objectives and challenges that motivate this feature of the flexible networking system. Returning to the design blueprint introduced in Chapter 4, the overall approach to the design of this component is then laid out. The blueprint influences the subsequent layout of the remainder of the chapter.

As one of the design aims of the flexible multi-protocol network-layer is to be as sympathetic to the underlying fundamentals of the primary network-layer routing components as possible, the design approach eschews, in so far as it is possible, the use of techniques that induce extra signalling costs. In this context, it is important to understand and characterise the degree to which an ad hoc node can appreciate the configuration of its local environment using the signalling generated by the primary network-layer components; such

matters are discussed in Section 6.3. This discussion continues the theme that was developed in the last chapter.

A detailed review and analysis of a number of distributed computing techniques that may inform the design of such a component is then presented in Section 6.4. Many of the observations noted in this review are generally applicable to the reconfiguration component described in the next chapter. Covering ad hoc management schemes, address auto-configuration and self-stabilisation techniques, this review helps to highlight the limits of some approaches and the possible application of other approaches within the confines of the networking system described in Chapter 4.

Finally, Section 6.5 presents the design of a protocol that addresses the problems that are associated with network-layer routing protocol auto-configuration. The auto-configuration protocol is composed of two sub-protocols; the Stateless Configuration Initialisation protocol and the Configuration Conflict Detection and Resolution protocol. As the design of the second protocol is dependent on the self-stabilisation techniques of local checking and local correction, a short primer on the specifics of these techniques is also presented in Appendix B as self-stabilisation falls beyond the general scope of networking technologies and concepts.

6.2 Auto-Configuration: Initialisation, Join, Migration & Merging

6.2.1 Objectives

The primary objective of this component is to allow nodes to dynamically auto-configure their network-layers with respect to their choice of routing protocol so that the state of the network, when viewed as the union of the states of its constituent nodes, is consistent and stable regardless of the naturally occurring disruptive events that occur. The proposed component should be flexible and robust enough to cope with the naturally occurring network events of node initialisation, node join, node migration, network partitions and network mergers. It should also be extensible so that other protocols may be accommodated in future systems and it should, naturally, incur as little overhead as possible. Essentially, it should ensure that a stable multi-protocol configuration prevails throughout the network.

6.2.2 Challenges

The design of an auto-configuration component is challenged by every attribute of the proposed ad hoc network; specifically, it is challenged by the open, autonomous and distributed nature of the system, as well as the lossy and often asymmetric links. It may be noted that a single-protocol network-layer, typically cited in the literature, would not introduce such complex network-layer configuration issues; it is the multi-protocol nature of the proposed network-layer, coupled with the aforementioned attributes of a realistic ad hoc network, which challenges the design of the auto-configuration component.

As described earlier in Chapter 3, with regard to Figures 3.10 and 3.11, a network of flexible multi-protocol nodes may be composed of, or decomposed to, a patchwork of heterogeneous regions of nodes, operating

their own choice of routing protocol, that reflect the varying underlying dynamics of the network. As ad hoc networks are transient in nature, especially when compared to more traditional infrastructure-based networks, it should be expected that networks may dissolve or reform after a time. Individual standalone networks may never encounter another standalone network in the course of their existence, or alternatively, multiple standalone networks may merge if the mobility of the constituent nodes causes multiple networks to occupy the same physical space. In essence, the networking landscape may feature multiple networks and multiple isolated, i.e. non-networked, nodes at any time. The challenges posed by common and naturally occurring mobile ad hoc network events are briefly noted here to bring the objectives into clearer focus.

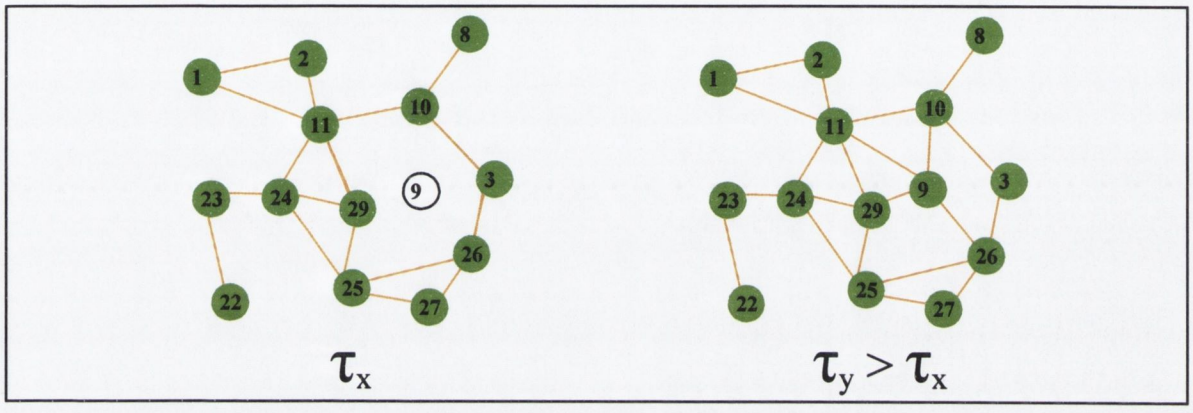


Figure 6-1 A Single Node Boots-up and Auto-Configures Itself by Joining the Network

A single node may be powered-on or booted-up in any region at any time, as depicted in Figure 6.1 at time τ_x. The region may be occupied by a network that is configured in one of the possible protocol states. The challenge for the auto-configuration component is to ensure that Node 9 determines the network configuration in a low-cost way and configures itself by τ_y, so that it may join the network.

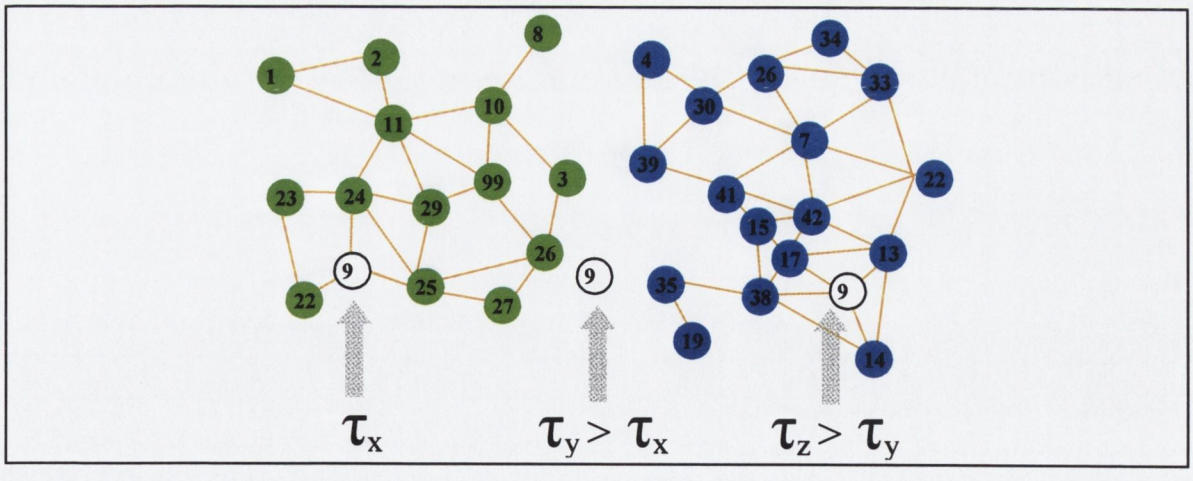


Figure 6-2 Node Migration between Two Standalone Networks

When a node is up and running, as depicted for Node 9 in Figure 6.2 at time τ_x, it may physically move from the region occupied by the green nodes to the region occupied by the blue nodes at time τ_z. The challenge for the auto-configuration component in this scenario is to enable the node to detect that it has moved from one region to another, or as depicted at time τ_y, has moved to a position between both regions, so that it can

configure itself appropriately. The component is challenged by the signalling nature of the nodes in the green and blue regions, as well as the signalling nature of Node 9 itself. Nodes may glean information about the surrounding network composition through the normal signalling generated by nodes in the course of operating their choice of routing protocol. Consequently, the migrating node may move from an active signalling area to an inactive signalling area, or the node itself may be actively signalling or inactive. Such signalling characteristics impact the manner in which the migrating node, Node 9, and receiving network, the blue region, handle the migration event.

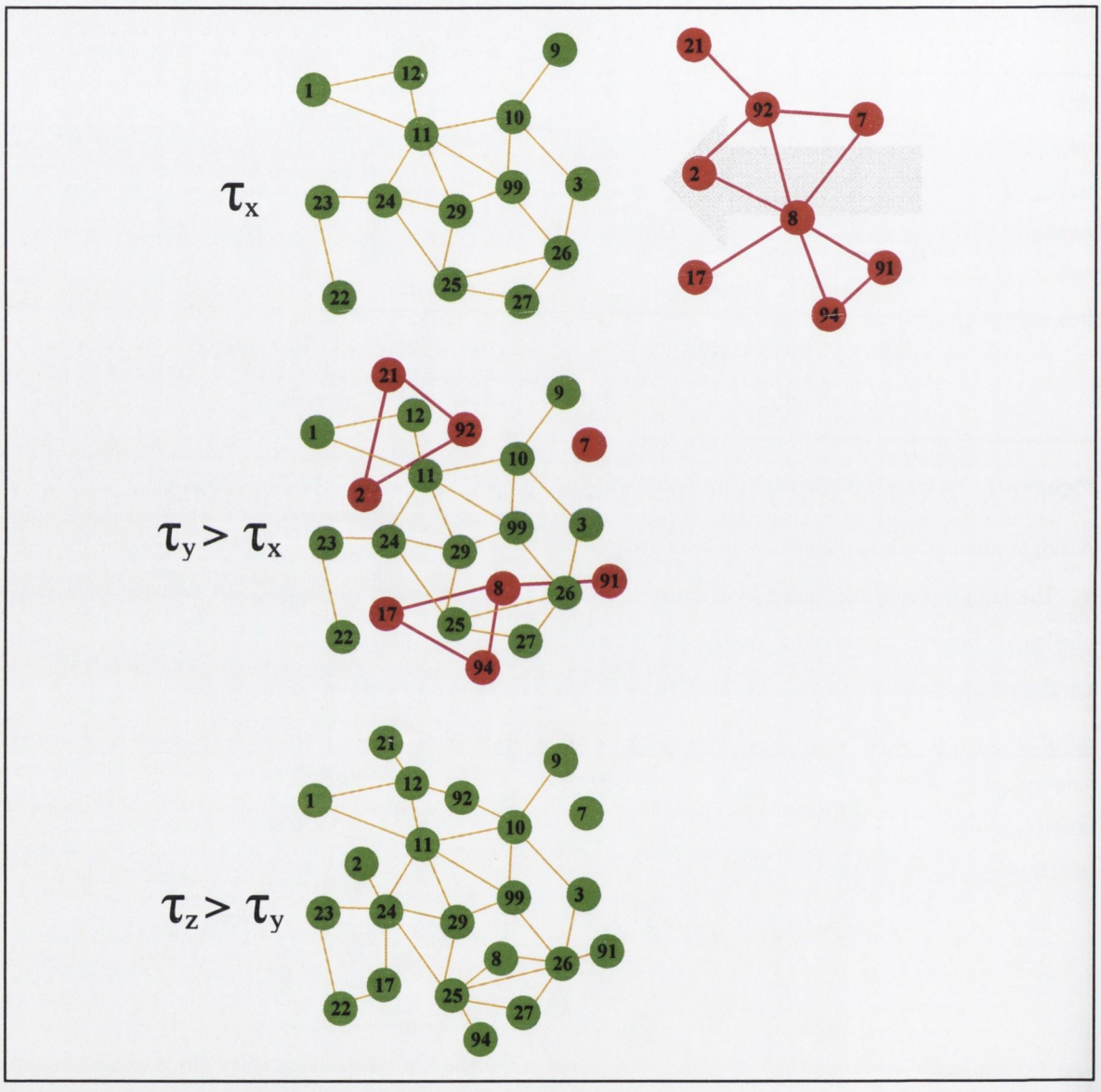


Figure 6-3 Two Standalone Networks Merge

Somewhat similarly to the migration scenario depicted in Figure 6.2, the nodes may also experience network mergers when two alternatively configured regions of nodes come to occupy the same physical space and ether. The absolute physical movement of the nodes is not relevant; only relative node movement affects the operation of ad hoc routing protocols. Figure 6.3 depicts a green region and a red region of nodes moving toward each other, relatively, at time τ_x. By time τ_y, the nodes have come to occupy the same space, but they are partitioned at the network-layer. Unlike the migration or join events, where an isolated node is received

by a network, in a merger event two functioning networks must be combined. Again, each network will have its own signalling characteristics which affect the amount of information that the two networks glean about the merger event. The challenge for the auto-configuration component is to enable both networks to homogenise their choice of routing protocol within a bounded time at least cost to the packet-forwarding operation of the network-layer.

6.2.3 Design Blueprint

Returning to the blueprint of the flexible multi-protocol network-layer which was introduced in Chapter 4, the design challenges for this component can be summarised as finding mechanisms that enable nodes to configure their choice of routing protocol so that they may form a network with their neighbouring nodes based on observations made possible by ambient signalling. The main elements of the component are highlighted in Figure 6.4.

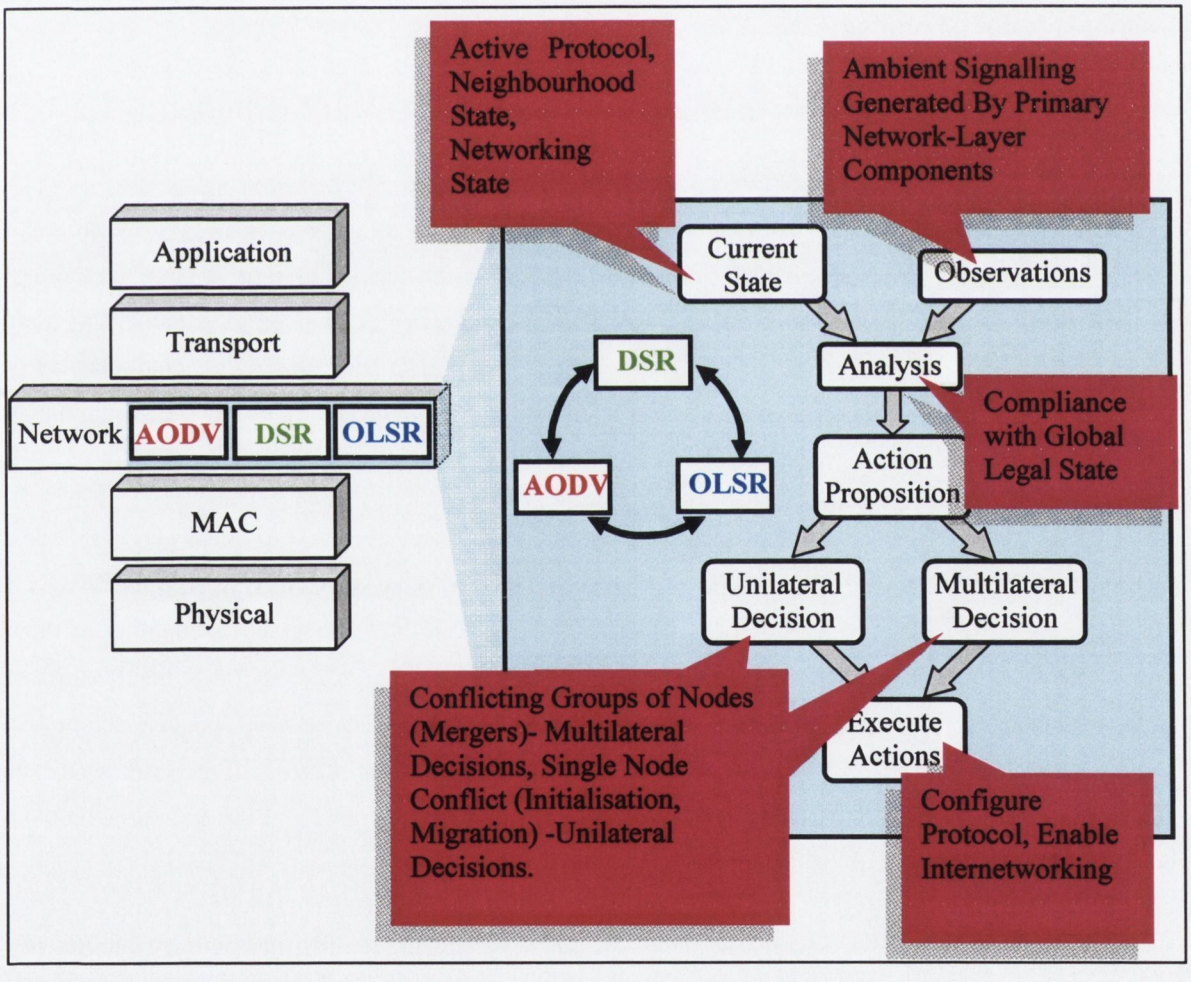


Figure 6-4 Blueprint for Design of Auto-Configuration Component

The actions taken by a node will be informed by its current state and its network observations. A node's state is defined by its current routing protocol configuration, whether or not it is actively networking with other nodes and the state of its links to neighbouring nodes. The node may observe the network by leveraging the information carried in packets that are issued in the course of normal networking, i.e. it may extract information from the packet headers of routing packets and data packets issued by neighbouring nodes. The

absence of such packets provides equally meaningful information. The effect of a node's current state on its ability to passively observe its networking environment by means of ambient signalling is discussed in the next section.

The analysis of the observations made possible by way of the ambient signalling should enable a node to detect whether it is in a legal state, i.e. whether or not it is operating the correct protocol, or whether it should trigger the node to take remedial action. Any such analysis will be predicated on the identification of what constitutes a legal global network configuration for nodes in a flexible multi-protocol ad hoc networking system. If the node detects that it is in an illegal state it must make decisions, whether unilateral or multilateral, to correct the network state with respect to the desired network configuration. As this component is tasked with resolving routing protocol configuration conflicts between nodes brought about by naturally occurring network events such as node migration, network mergers and node initialisation, the techniques that will be used must allow for situations in which both conflicting groups of nodes must resolve their configuration differences (multilateral decisions) and situations in which a single node is out of step with the network (unilateral decisions).

6.3 A Node's Perspective through the Lens of Ambient Signalling: Network Configuration

As noted in the previous section, the design of a network-layer configuration component revolves around a node's ability to make pertinent observations about the environment in which it is operating. A node's ability to make such observations, based on ambient signalling techniques, is determined by a number of node characteristics. This section presents a discussion of such characteristics and describes the consequences of various combinations of such characteristics on a node's ability to appreciate its networking environment from the point-of-view of the network configuration.

As discussed in Chapter 5, the level of activity on the ether surrounding a node determines a node's awareness of its local networking environment. From the auto-configuration component's perspective, it is the configuration of the neighbouring nodes which is important, and not the density or mobility of those nodes. The signalling generated by the network-layer, either through route discovery, route maintenance or packet-switching, informs nodes as to the nodal composition of their neighbourhood. A node's perspective on the local composition of the network may be characterised by a number of network-layer and physical-layer properties; the signalling activity of the network-layer, the quality of the node-to-node radio links and the connectedness of the node.

Nodes may be generally classified as either active or inactive with regard to their signalling profile. Actively signalling nodes are nodes that issue packets on a regular basis, i.e. with enough frequency to suit a node's routing protocol so that it may discover and maintain connections with neighbouring nodes; the specific minimum intervals vary among the protocol specifications. Proactive nodes, i.e. nodes operating table-driven protocols such as DSDV or OLSR, continually issue packets to maintain up-to-date topology information regardless of the traffic demands on the network. Nodes operating on-demand protocols such as AODV and DSR issue packets when they are attempting to discover and maintain routes and when they are forwarding

packets. However, when traffic falls off, nodes operating reactive protocols become quiet, i.e. they become inactive from a signalling point-of-view. Regions of nodes may be characterised by this signalling characteristic; an inactive region may consist of quiet DSR nodes which, after a time corresponding to their route cache timeouts, will expunge any information about their neighbourhood if no traffic is applied to the region. This characteristic of a node or a region of nodes has a bearing on the manner in which nodes may notice and react to inconsistencies in the state of the network-layer configuration over a region of nodes.

Furthermore, node-to-node links may be characterised by their quality, i.e. whether or not the links are symmetric or asymmetric. This link property is determined by the underlying radio properties intervening between the transmitting and receiving nodes. (Asymmetric and symmetric links may also be referred to as unidirectional and bidirectional links, respectively.) A symmetric link is one in which both nodes on either side of a link receive the transmissions from the other side. An asymmetric link is one in which only one node receives transmissions from the other node. Asymmetric links may occur when the transmission range of one node is less than the transmission range of the neighbouring node, as depicted in Figure 6.5. The dashed pink rings denote the extent of each node's transmission radius. For any given node, the link between it and an adjacent node may be asymmetric. The bidirectional link between any pair of nodes may be considered to be composed of two asymmetric (or unidirectional or anti-symmetric) links; one from node *u* to node *v*, the other from node *v* to node *u*. Using this link model, it may be said that each node has an incoming link from its neighbour and outgoing link to its neighbour. If the link between the two nodes is asymmetric, then it may be the node's incoming link or its outgoing link that is down. In Figure 6.5a, node *u*'s outgoing link to *v* is not up, but it can receive a signal from node *v*. Figure 6.5b shows the contrary case where node *v* is aware of node *u* as node u's outgoing link, i.e. node *v*'s incoming link, is up and working.

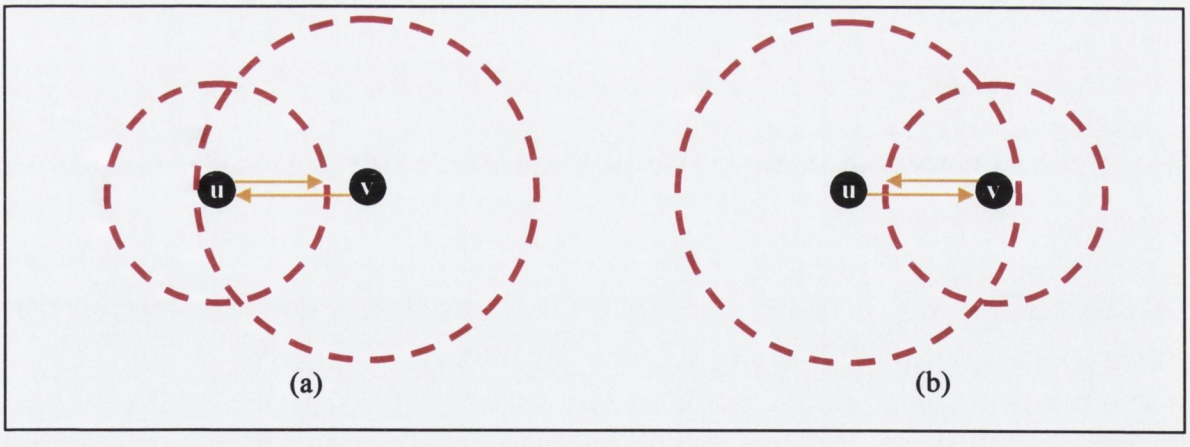


Figure 6-5 Link Asymmetry may Occur in the Incoming or Outgoing Direction

Such asymmetric links are accommodated in each of the candidate routing protocols. DSR and AODV explicitly recognise and accommodate the existence of asymmetric links, allowing for the discovery of routes that may follow asymmetric paths from end-to-end, either partially or completely. While OLSR relies on symmetric links to construct its optimal multi-point relay set of repeater nodes, it also accommodates asymmetric links which are advertised in its topology update messages.

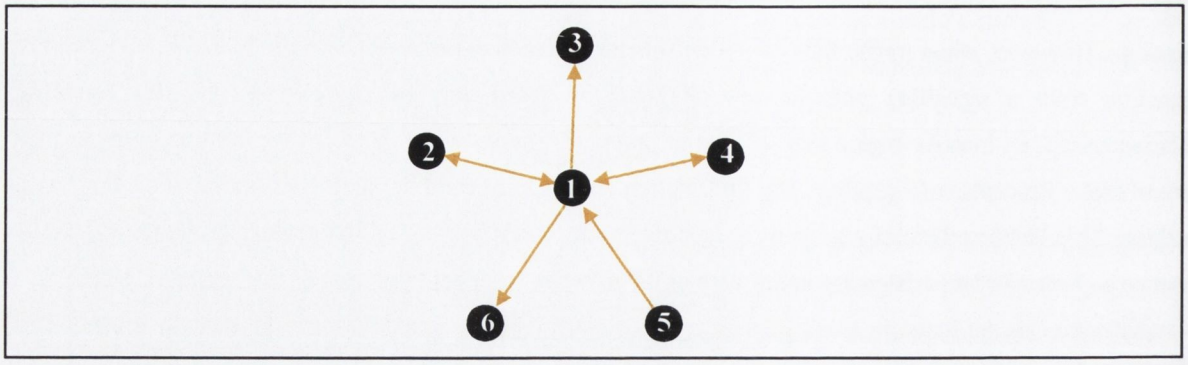


Figure 6-6 Non-Mutuality of Neighbourliness

As a consequence of a node's level of signalling activity and the possible presence of asymmetric links, a node will not always have a full view of its neighbourhood if it can only rely on locally broadcast messages. A node can only *know* of nodes that it *receives* messages from. In the case where a node's incoming link is down, the node cannot independently infer whether its outgoing link to another node is up or down. It may only learn of the status of this link if the information is purposefully diffused via another neighbour, a method employed by the three routing protocols in the course of discovering the network topology. However, such diffusion techniques amount to multi-hop route discovery and are unsuitable for use in an auto-configuration component as they replicate the services of the underlying routing protocols. Consequently, the auto-configuration component should only be reliant on locally broadcast information that does not need to be relayed through intermediate nodes. The possibility of asymmetric link quality means that local state knowledge can vary from node to node; in essence it means that the neighbourliness of two nodes is not always mutual. An example of such a situation is depicted in Figure 6.6. While Node 1 is a neighbour of Nodes 2, 3, 4 and 6, as those nodes can receive messages from Node 1, only Nodes 2, 4 and 5 are neighbours of Node 1. The impact of the non-mutuality of neighbourliness impacts the techniques that can be relied upon in designing robust network protocols.

Bearing the activity of the nodes and the quality of the links between them in mind, it is possible to categorise nodes as being either *isolated* from the network or *connected* to the network. As nodes are autonomous with respect to their mobility, they may move away from the area occupied by the network that they had joined, leaving them physically isolated from their network. Alternatively, a node may not become physically removed from its network but it may experience physical-layer events that cause it to lose incoming links from its neighbours, thereby also rendering it isolated from its network.

In short, an isolated node is one that has lost contact with its network regardless of the underlying causes. A node can easily establish the fact of its isolation by querying its Neighbourhood Table. If the node has no recorded neighbours, then it may consider itself to be isolated. Referring to Figure 6.7, after the process of initialisation Nodes 1, 2, 3, 4, 5 and 6 operate the DSR protocol. Node 1 is within transmission range of five other DSR-configured nodes. The dashed pink rings denote the extent of each node's transmission radius. As no traffic has been applied to this region of the network since it was initialised, these nodes are inactive with regard to signalling. While Node 1 is within transmission range of every other node, and they are within transmission range of it, Node 1 is not aware of the presence of those neighbouring nodes. In the absence of such neighbourhood knowledge, Node 1 considers itself to be isolated. Indeed, an inactive network region,

such as the region composed of Nodes 1, 2, 3, 4, 5 and 6, consists a group of isolated nodes; while they may be part of the same network, they are not aware of each other as they are not connected.

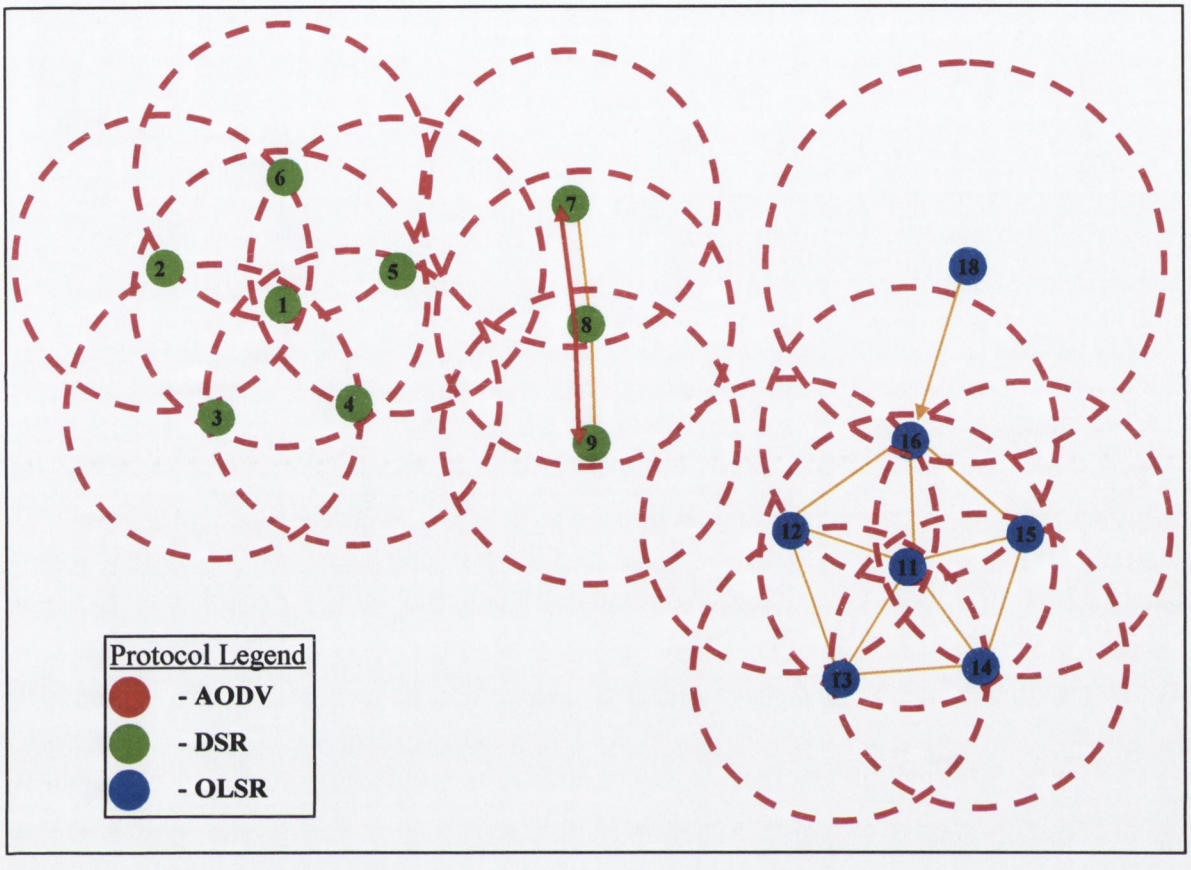


Figure 6-7 Illustration of Isolated and Connected DSR and OLSR Nodes

In another section of the network, Node 18 operates OLSR. As this is a proactive protocol, Node 18 repeatedly issues packets. It may be seen that Node 16, also operating OLSR, is within transmission reach of Node 18. While Node 16 is aware of Node 18's existence, as a result of the asymmetric link between Node 16 and Node 18, Node 18 cannot learn of Node 16's existence. Node 18, while actively signalling, may consider itself to be isolated from the network.

On the other hand, a connected node is one which is part of a network and is actively connected to at least one other node in that network. A node may simply detect whether it is connected to its network by querying its Neighbourhood Table. Nodes 11, 12, 13, 14, 15 and 16 are operating OLSR and as they are all within transmission range of each other and are actively signalling, these nodes are aware of their neighbours. Also, Nodes 7, 8 and 9, operating DSR and experiencing a flow of traffic, are also connected as the signalling activity causes them to become aware of each other.

To summarise the effects of these network-layer and physical-layer characteristics on a node's local network perspective, it is helpful to review Figure 6.8 which depicts a variety of nodes experiencing the various aforementioned characteristics and to consider the effects of these characteristics on the nodes' ability to auto-configure themselves in the face of such naturally occurring network events. The dashed orange lines indicate possible node-to-node links which may be established between nodes if they were to issue packets;

the unbroken orange lines indicate symmetric links which have been established between nodes through signalling.

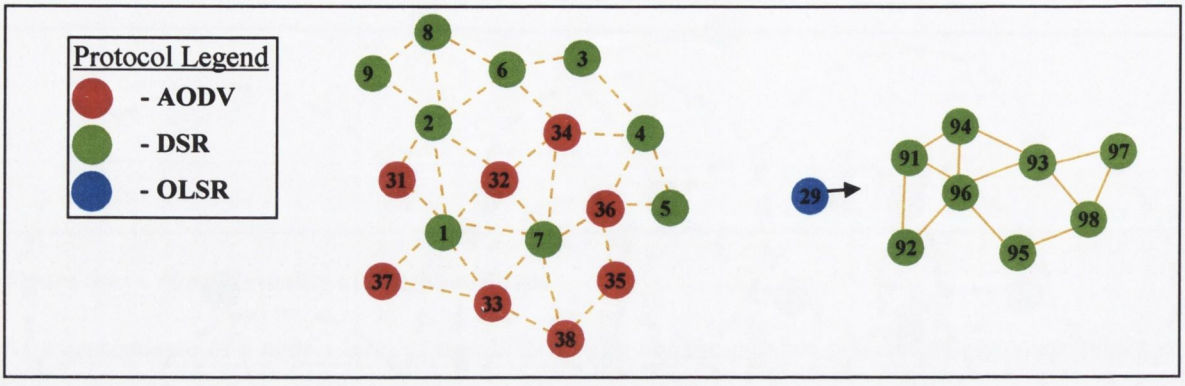


Figure 6-8 Node Isolation and Signalling Activity: The Impact on Reaction to Network Events

In the first case, two groups of nodes occupy the same physical area and ether to the left of Figure 6.8. One group of nodes is configured to use AODV, the other DSR. No traffic is applied to this region of the network at this time. As a consequence of the lack of service demand from higher layers, all of the reactive nodes in this region are silent which will leave them isolated after a time. The absence of signalling in this region means that Node 7 is unaware of the (physical or geographical) neighbouring presence of Nodes 1, 32, 34, 36, 35, 38 and 33 which are within transmission range of it. While Node 7's immediate *possible* neighbourhood consists of a mix of both AODV and DSR nodes, the configuration of these nodes has no relevance to Node 7 as it has not attempted to connect to the network owing to a lack of demand for its services from higher layers. As such, as an inactive node, in the presence of other inactive nodes, there is no conflict between Node 7 and its neighbour's configurations. Problems would only arise if such a node, or nodes neighbouring it, became active in which case the affected nodes would have to deal with a merger event.

In the second case, an active but isolated node, Node 29, is moving towards the active region of DSR nodes to the right of Figure 6.8. As the region of DSR nodes is active, these nodes are connected to each other and are somewhat aware of each other's presence, in so far as the node-to-node link quality allows. As Node 29 encroaches on the DSR region both it and the nodes that it impinges upon will detect each other due to the signalling being generated by the routing protocol or packet-switching components of their network-layers. Such signalling, together with both Node 29's and the DSR region's awareness of their connectivity status should enable them to resolve the configuration conflict by means of Node 29 adopting the DSR protocol.

While the reliance on signalling generated by the primary network-layer components means that nodes may not be configured 'correctly' from a network-wide point-of-view at all times, i.e. the region of the network to the left of Figure 6.8 would remain inconsistent due to a lack of awareness of the situation, this does not lead to networking problems as such inconsistencies are permissible in the absence of network-layer service demand in that region of the network. By utilising the signalling information that is generated by the primary network-layer components, the corrective actions of the auto-configuration component are directly triggered by the needs of primary network-layer components. However, in order that the auto-configuration component

does not impede the network-layer's multi-hop ad hoc routing service it must act in a fast and efficient manner once it detects inconsistencies in the state of the network's configuration.

In the absence of network-layer activity there is no need for a node to actively pursue corrective action. However, when network-layer activity restarts, the information needed to auto-configure the network is provided by the signalling generated by the primary network-layer components.

6.4 Techniques of Interest

Having made observations about the composition of the network in which the node is operating, the node must be enabled to act on such observations in a collaborative manner such that its decisions, whether unilateral or multilateral, cause the network to converge to a desired state.

In order to inform this aspect of the design of configuration components for the flexible multi-protocol network-layer system, it has been found necessary to consider a number of techniques, systems and concepts that already exist in the literature. Many of the observations noted in this review are generally applicable to the reconfiguration component described in the next chapter. The techniques that are surveyed include address and router configuration in ad hoc networks, self-stabilisation and ad hoc network management schemes. While these techniques do not exhaust the body of possible sources of interest, they have been chosen because they are representative of a number of different approaches to the issues surrounding the distributed organisation of systems. Prior work in the area of techniques that enable network-layer flexibility of the kind proposed and discussed in this chapter and in Chapter 3 is *very* limited.

The fundamentals of each technique are introduced and overviewed in the following subsections. These techniques are analysed with respect to their suitability for use in a flexible multi-protocol ad hoc networking system. This review helps to highlight different types of fundamental distributed design approaches that may or may not be extended to the mobile ad hoc networking domain.

6.4.1 Ad Hoc Network Management

Ad hoc network management schemes such as the Guerrilla scheme [She03], the Ad Hoc Management Protocol [Che99], [Gur98] and [Kan01] have been developed specifically for ad hoc networks. They are used to manage networks in response to perceived changes in the state of the network. While, at first glance, an ad hoc network management scheme may seem like a suitable technique to address all of the problems associated with the enabling of flexibility at the network-layer of ad hoc networks, closer analysis reveals that the schemes used are too reliant on centralised and hierarchical structures.

Such management schemes are modified versions of the Simple Network Management Protocol (SNMP) [Cas90], the aim of which is to provide fault, performance and configuration management to TCP/IP-based networks [Aid94]. Fault management deals with the detection, isolation and correction of faults in network entities whilst configuration management is generally charged with the initialisation and shutdown of a network. Configuration management also handles the configuring of network entity parameters to provide differentiated services, the addition of new entities to a network and the upgrading of existing entities.

Performance management involves the collection of network statistics and the tuning of the network parameters to improve performance. In order to provide these services, network management protocols must frequently observe the state of the network. The observed information is used to infer the QoS in the network relating to faults, congestion and attacks.

The Guerrilla management scheme for ad hoc networks is based on a three-tier (supervisor / nomadic manager / probe) system. A supervisor regulates and distributes management polices to nomadic managers, which in turn use active probes to locally manage ordinary nodes. The nomadic managers can be either manually selected or can be elected by way of a clustering algorithm that creates trees of various depths. Such trees are rooted at the manager. The probes actively sense the network dynamics when they are transmitted in data packets to remote nodes. Monitoring probes may be transmitted multiple hops away to gather information on the neighbouring topology. The probes report back to their nomadic managers and data may be aggregated and sent further upstream for more centralised consideration. Further management probes are then disseminated to adapt the network in order to address perceived changes. The scheme presumes the existence of a functioning proactive multi-hop routing mechanism as the probes, nomadic managers and the supervisor may be multiple hops away from each other.

The ANMP scheme is similar to the Guerrilla scheme, using a three-tier structure to scale the management process. Again, the hierarchical structure is supported by a clustering algorithm. The authors argue that a hierarchical model is appropriate for data collection and propose two clustering algorithms. The first is based on GPS information while the second algorithm uses graph theory to generate the clusters. The algorithms presuppose the existence of a functioning routing protocol. The clustering algorithms are used to divide the network into logical partitions for ease of management at a higher level. The authors of ANMP view ad hoc networks as an extension of the wired static networks for which SNMP was designed and do not believe that SNMP-like architectures are incompatible with the ad hoc networking domain. As such, ANMP has been designed to be SNMP-compatible. Charles E. Perkins, on the other hand, believes that, in the context of mobile ad hoc networks, SNMP-type architectures are "*completely off the map*" [Per01]. Application-layer management schemes require the support of too many lower level services, such as peer-to-peer multi-hop routing and clustering, to be contemplated for adaptation for use at the network-layer.

Not withstanding the merits of ad hoc network management schemes with regard to network organisation, it is apparent that they conflict with the stated assumptions underlying the design of the flexible system which were enumerated in Chapter 4. As a consequence, ad hoc management schemes will not be analysed further in relation to their suitability to enable the flexible components identified in Chapter 3.

6.4.2 Address and Router Auto-Configuration

Address and *router auto-configuration* is an area of research within the ad hoc networking community that covers the investigation of mechanisms that allow nodes (routers) to automatically configure certain variable parameters that are necessary for network participation. At this stage, it is perhaps wise to note the differences between the network-layer configuration presented in this literature and the network-layer configuration proposed in this thesis. The aim of the network-layer configuration in this literature is to ensure

that each node is uniquely identifiable, whereas the aim of network-layer configuration in this thesis is to ensure that multi-protocol nodes cooperate to employ a suitable ad hoc routing protocol. Addresses and routing protocols are both *variable* and *dependent* network parameters. They are variable in that a transient, autonomous, self-configuring mobile ad hoc networking node may need to adapt these parameters in response to the changing network scenarios. They are dependent parameters as a node's choice of address and routing protocol must be compatible with the choices being made by the other participating nodes. Nonetheless, it is possible to make a number of observations about the processes used in this literature that may apply to the flexible components proposed in this thesis.

As conventional approaches, such as the Dynamic Host Configuration Protocol (DHCP) [Dro97] or manual configuration, are not suitable for mobile ad hoc networks, the issues surrounding the automatic configuration of ad hoc routers have received some attention in [Bol02b], [Mca00], [Nes02], [Par04], [Sun03b], [Shi04], [Ton03] and [Vai02]. Thorough reviews of this literature may be found in [Sun04] and [Wen04]. McCauley *et al.* [Mca00] have developed a Dynamic Registration and Configuration Protocol (DRCP), which adapts DHCP for the ad hoc networking domain, together with a Dynamic Address Allocation Protocol (DAAP). The authors aim to provide a flexible and robust router registration and address auto-configuration system for quasi-dynamic networks; networks that fall somewhere between mobile ad hoc networks and more stable back-boned networks, where only the peripheral nodes are mobile. They argue that the DHCP model is a satisfactory model to extend into this domain. As the network is 'quasi-dynamic', and not a truly mobile ad hoc network, the scheme does not seek to address the problems caused by network partitions and mergers.

Most of the other approaches that have been proposed for use in mobile ad hoc networks disagree with the contentions of [Mca00]. The rest of the literature recognises network events such as node initialisations, node joins, node departs and network mergers as events that are typical of a transitory ad hoc network. As they are not exceptional events, they should be dealt with by tailor-made auto-configuration mechanisms rather than modified DHCP-based schemes. One such approach is the MANETconf [Nes02] scheme, which presents a distributed dynamic host configuration protocol for nodes in a mobile ad hoc network. The authors of MANETconf have modified the Ricart-Agrawala [Ric81] mutual exclusion algorithm for use in the ad hoc networking context, removing the original algorithm's constraints on bounded message delays, reliable multicasting, fixed network size and topology changes. Each and every node keeps a copy of the list of allocated addresses rather than relying on a single central server to hold this information. It is argued that a mutual exclusion algorithm is required to control access to the network's address pool in the absence of a central server. New nodes wishing to join the network choose an already configured node as an agent to do their bidding while a *soft-state* address is assigned to them. The soft-state approach allows a node to take an address while the allocation process is ongoing. If the attempted allocation is denied to the node, or if the node does not receive a confirmation message within a fixed period, then the node relinquishes the address. This approach guarantees termination of the process in the presence of lost or delayed confirmation messages. MANETconf may be described as a *stateful* scheme, meaning that each and every node in the network knows the exact global network configuration with regard to address allocation. As the scheme is stateful, in a network that may experience many mergers, partitions and joins, the overhead incurred during

the exchange of state information between nodes is not insignificant in the context of the constrained bandwidth of a wireless ad hoc network.

Boleng presents the more radical of the approaches, challenging the status quo with regard to the addressing structure [Bol02b]. Existing schemes assume the use of conventional IP-like addresses and address blocks. This scheme enables nodes to use a minimal addressing structure that adapts to the changing size of the network. The variable-length addressing scheme is motivated by the authors' desire to reduce control overhead by using the minimum number of bits possible to address every network node uniquely, e.g. a network of up to 16 nodes would use 4 bit addresses whereas a network of up to 65,536 hosts would use 16 bit addresses. As the address size reflects the size of the network, it must be changed when nodes join (or leave) the network or if two standalone networks merge. Nodes are allocated sequential addresses, i.e. when a node joins the network it receives the next-highest address available in the address block. The means by which this scheme enables node joins is very similar to the MANETconf scheme, involving the use of a bidding agent and communication with every other node to inform them of the arrival of the new node and the allocation of the address. The authors acknowledge that this type of procedure introduces a considerable amount of overhead. Boleng also modifies the network-layer packet header so that two extra fields are created to indicate the address length being used in the network and also the highest address that has been allocated. This information amounts to global state information that must be kept consistent across the network at all times. The variable-length address approach adds extra complexity to the traditional problems associated with address auto-configuration: changing the variable address length requires all nodes to take action during runtime, i.e. '*on the fly*'.

Another interesting feature of Boleng's scheme is the way in which it allows nodes of one network partition to detect nodes of another partition using the address state information, i.e. the 2-tuple of the current highest address and address size carried in the network-layer packet header. The 2-tuple values held by neighbouring nodes should be relatively close in value, i.e. the address size and the highest current address should not vary wildly from neighbour to neighbour. If a node detects a neighbouring tuple value that is beyond some threshold, then it may assume it has come into contact with a foreign network partition. The use of this approach means that the network does not have to manage a separate network identification scheme. While Boleng has introduced extra complexity into the addressing scheme by using a variable address structure, this approach has lead the author to consider more issues than are normally prevalent in address auto-configuration schemes, namely the reconfiguration and simplification of the global state information.

While the MANETconf and Boleng schemes provide strong network consistency, the proposal presented by Vaidya introduces the concept of weak consistency [Vai02]. Strong network consistency aims to ensure that all nodes are always fully consistent with regard to the desired network state, i.e. each node has a unique address at all times. Weak consistency, on the other hand, relaxes the demand that all nodes should be consistent at all times and introduces techniques that allow the network to mitigate the possible negative effects due to possible inconsistencies. In contrast to the stateful configuration approach of MANETconf and Boleng, [Vai02] proposes the use of a stateless address configuration algorithm. The use of stateless address mechanisms, when a node tentatively picks its address based on local knowledge such as its MAC address, generally requires that a separate mechanism is used to resolve address conflicts in the network, e.g. two

nodes using the same address. Such schemes are referred to as duplicate address detection algorithms, which themselves may be weak or strong. Strong duplicate address detection schemes guarantee that there will be no duplicate addresses within the network whereas the goal of a weak duplicate address detection scheme is to prevent a packet from being routed to the 'wrong' destination address in the event of duplicates existing in the network.

Vaidya proposes the use of a weak duplicate detection scheme primarily because the strong schemes are reliant on the use of timeouts. Previous schemes such as those presented in [Bol02], [Per00] and [Nes02] used timeouts to account for lost or delayed messages. However, Vaidya argues that ad hoc networks will often be subject to unbounded delays and has shown that if even one link between two nodes is unbounded, then a timeout-based strong duplicate address detection algorithm may fail. This is a reflection of the distributed consensus impossibility result which will be discussed in the next chapter. Unbounded delays may occur when nodes leave the network permanently or when the network partitions. It is not practical to place a bound on such delays, which would be manifested by the use of a large timeout, when such delays are very likely to occur. The delays brought about by lengthy timeouts would slow the convergence of the scheme. It is also noted that even if the delays were bounded, it is not trivial to determine the bound when the network size is unknown; this would be especially true of an open network.

The [Vai02] stateless address configuration mechanism bears some similarity to the IPv6 mechanisms that allow nodes to determine link-local addresses which have a limited internetworking scope. The weak duplicate address detection algorithm proposed by Vaidya involves modifications to the underlying routing protocols that allow for the detection of duplicate addresses in the course of normal network routing. Essentially, each node is expected to have a unique '*key*', e.g. a MAC address or randomly chosen number, in addition to its network-layer address. In effect, each node has a dual identity. The uniqueness of these keys is to be assured by use of a separate duplicate address detection algorithm. The approach is similar to IPv6, except that instead of embedding the key (e.g. a MAC address) into the network-layer address, the key is embedded in the underlying routing protocol headers instead.

Finally, Sun *et al.* comprehensively describe a scheme that enables dynamic, distributed address configuration [Sun03b]. The scheme operates in multiple stages, building primarily on an earlier address auto-configuration protocol [Per00]. The scheme involves address generation, address duplicate detection and address registration with an Address Authority. It is perhaps the most mature and comprehensive solution to this problem yet published, advancing on the work presented in the publications previously cited in this section. The authors accept the contention of Vaidya, that unbounded message delay renders strong duplicate address detection impossible. However, they argue that in a well-connected network it is possible to bound delays by choosing appropriate timeouts. Essentially, the scheme overcomes the problems of the seeming impossibility of strong consistency by operating on the premise that it is possible to have strong consistency in a well-connected network partition and that it is possible to successfully merge two such partitions. Sun *et al.* foresee well-connected nodes organising themselves using the combination of a weak distributed request/reply algorithm that is supplemented by two centralised back-up entities. The use of such centralised back-up entities requires the use of multi-hop routing services. Nodes that are not well-connected, e.g. nodes at the periphery of the network, nodes that do not have steady links and nodes that leave or join the

network sporadically, are accommodated by developing other strong mechanisms that allow them to integrate with the network when they become well-connected again.

In summary, the benefit of stateful schemes is that they leave no room for ambiguity about the state of the network. MANETconf, Boleng, and to an extent Sun *et al.*, aim to ensure that all the nodes are strongly consistent at all times. Multi-hop routing protocols and higher-level services and applications may function seamlessly over the network as they would if the nodes were using pre-assigned static addresses. However, the MANETconf and Boleng schemes incur much overhead with regard to message passing and state tables. Both schemes require initialising nodes to proactively negotiate with the entire network using positive confirmation techniques. Contrary to the stateful approaches, stateless schemes absolve the network of the requirement to maintain consistent global state information, but in so doing they require the implementation of mechanisms that mitigate against state inconsistencies. The Vaidya scheme proposed a weak duplicate address detection scheme which essentially tackles problems that may arise from the existence of duplicate addresses by using an alternate addressing scheme in parallel with the main scheme. However, the benefits of such a low-cost stateless configuration and weak duplicate address detection approach may be offset by the changes necessary to many elements of the pre-existing network-layer components.

6.4.3 Self-Stabilisation

Self-stabilisation (or stabilisation) is the branch of computing that focuses on the ability of a system to converge, within a finite number of steps, from an arbitrary state to a state that exhibits desired system behaviour. In other words, a protocol would be said to be self-stabilising if its specification does not require a particular initial state to be imposed on the system to ensure the correct behaviour of the protocol. Also, significantly, in contrast to a distributed consensus protocol that must be terminated in some way, a self-stabilising protocol is non-terminating. (Distributed consensus is reviewed in the next chapter.) The process of self-stabilisation continues as long as the system persists.

Self-stabilisation has been reviewed in the context of this thesis in order to inform the solution about means by which a distributed system may be strengthened in the face of disruptive events. In this thesis, such events may be manifested though the normal network events of migration, network partitions and mergers or network reconfiguration. A flexible multi-protocol system may be viewed as a multi-state distributed system that is subject to state perturbations through such events. The concept, or technique, of self-stabilisation is a general approach that may be applied to many distributed algorithms or protocols.

While the concept of a correct initial state makes sense in a single sequential program, such a concept in a distributed system of processes seems artificial. In the context of networking, a self-stabilising networking system may have an initial global state with arbitrary messages in its links, i.e. in outgoing or incoming queues, and arbitrary corruption in the protocol state variables of the network's nodes. A self-stabilising networking system would eventually converge to correct global behaviour. The concept was originally pioneered in 1973, when Edsger W. Dijkstra, already noted in the field of graph theory for his self-named shortest path algorithm, published a short paper that introduced the idea of self-stabilising systems in a distributed computing environment [Dij74]. Dijkstra introduced the concept in the context of stabilising a

distributed mutual exclusion algorithm on a ring (of processes). It took some years before the value of his contribution in that terse paper was further explored and developed. Leslie Lamport highlighted the work in a 1983 address to an Association for Computing Machinery (ACM) distributed computing conference, which led others to investigate the self-stabilisation approach to the development of robust fault-tolerant algorithms in distributed computing environments. Since that time, the model presented in [Dij74] has been expanded and its limits have been explored under varying system models, conditions and assumptions. Reviews of this field may be found in [Dol00] and [Sch93].

Jayaram *et al.* have shown that a crashing network protocol in an asynchronous, memory-less environment, in which messages may be lost, can be driven into an arbitrary global state [Jay96]. The authors considered network protocols like IP and High Level Data Link Control (HDLC) that do not require nodes to have non-volatile memory. After a crash and restart, whether due to power failure or system corruption, the protocol state variables are set to their initial values as the values at the time of crashing have been lost. While non-volatile memory is cheap and it may now be factored into protocol design, historically protocol designers have not relied upon the existence of non-volatile memory. Current ad hoc routing protocols such as AODV have been designed on the assumption that protocol state information may be lost when a node crashes and reboots. [Jay96] proves that system crashes, even ones as simple as node failures, may send networking systems into arbitrary states unless the protocols are self-stabilising.

Thus the self-stabilisation property is desirable in distributed networking systems that have to cope with unanticipated faults. The fault-tolerance that stabilisation brings to a protocol is often cited as the prime motivating factor for design of self-stabilising protocols. As noted, the configuration components proposed in Chapter 3 have a shared objective; to bring about a consistent state in the network so that remote nodes may communicate over multiple hops. It should be recalled that in a multimodal ad hoc networking system the global network state is the union of the local nodes' states. Inconsistencies in the global network state of a distributed system may be viewed as faults.

The applications of self-stabilising systems have been expanded from mutual exclusion algorithms to spanning tree construction algorithms, used for multicasting and clock synchronisation algorithms. Self-stabilisation applies to distributed systems which may be a networked group of processes or multiprocessor computers. The latter example may use a shared memory communication model which lacks any analogy with the ad hoc networking environment and analysis of, or observations on, this type of self-stabilisation system are not presented here.

The time complexity of self-stabilising systems varies. Clearly, a protocol that promises convergence to correct system behaviour is desirable, but not at a cost of a lengthy period of time during which the system may act inefficiently or not at all. The convergence time of a self-stabilising protocol should be such that the protocol manages to stabilise before new faults occur, otherwise a cycle of instability may exist. This would be unacceptable in a networking context. As was shown in Chapter 2, the performance of certain ad hoc routing protocols suffered when the protocol took too long to converge in the face of continual change in the network topology. Self-stabilising techniques that rely on centralised concepts and entities may be slow to converge and have a high message complexity. They may also be reliant on functioning multi-hop routing services.

Awerbuch *et al.* [Awe91] and George Varghese [Var92], in an attempt to address and simplify the issues surrounding the problem of convergence time and message complexity, refined the concept of self-stabilisation by introducing the notions of *local checking*, *local correction* and *counter flushing* among other concepts. As noted, in the networking context the global state of a network is defined as the local state of each node and link, i.e. the messages on the link passing between nodes. Simply and briefly put, the network model is represented in the usual manner by a symmetric, directed graph. Each link in the system may be viewed as an individual subsystem, as illustrated in Figure 6.9. For every pair of nodes, i.e. each link, there is an arbitrary leader, the selection of which may simply be a function of the two nodes' addresses or unique identifiers.

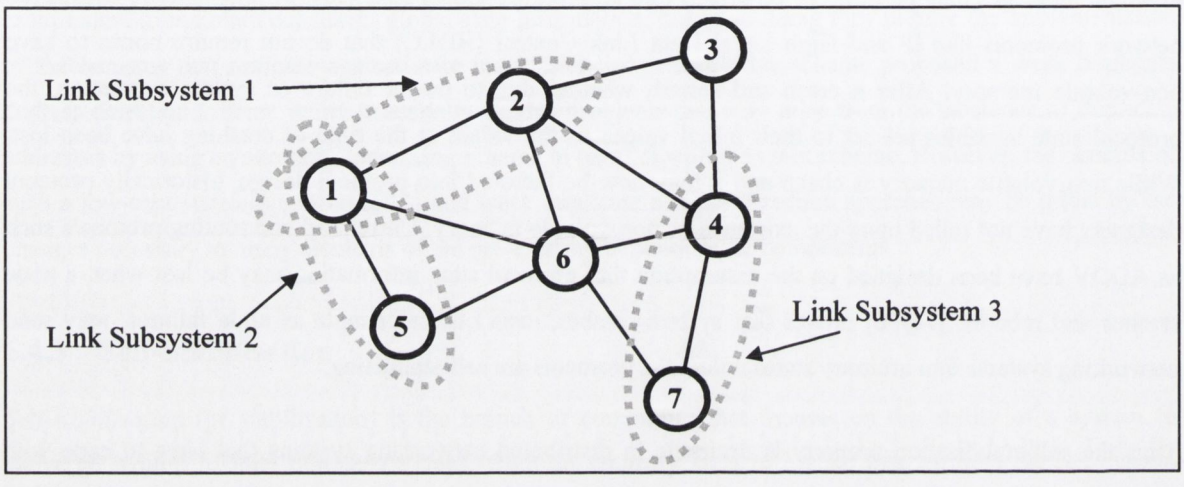


Figure 6-9 System Viewed as Union of Overlapping Subsystems

The main thrust of the idea behind local checking is that for many protocols it can be shown that whenever the protocol is in an illegal global state some link subsystem must also be in an illegal state. The efficiency of this approach compared to a global checking one is that state information does not have to be collated at a central node before it is checked for consistency. Each link subsystem can be checked in parallel. Local correction operates using a similar principle to that of local checking. Many protocols can be returned to a legal state by having each underlying subsystem independently correct itself to a legal state so long as the correction of one link does not invalidate another overlapping link.

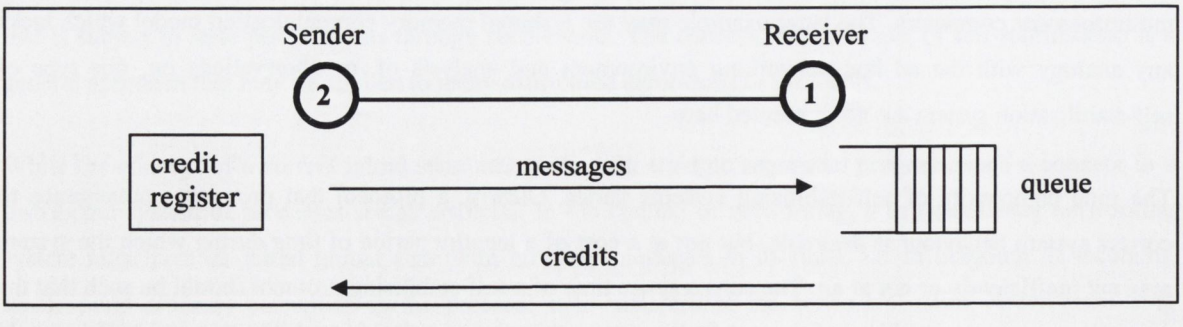


Figure 6-10 Varghese's Local Checking and Correction Example

Informally, the local checking and correction concepts may be explained by way of a simple example used in [Var92] which features two connected nodes, a simple network, that use a credit-based flow control protocol to throttle the flow of messages between them, depicted in Figure 6.10. The Sender node may send messages

to the Receiver node which buffers the messages in a queue of finite size. If the Receiver's queue is full, messages are dropped, so a credit scheme is used which allows the Receiver to indicate to the Sender the number of credits available. The Sender only sends messages when the credit register has a non-zero value, which it then decrements on sending a message. The Receiver sends the Sender a credit when it removes a message from its queue. Without further explanation of the detail of the scheme, it is apparent that the loss of messages or credits during the execution of the protocol or the crash/restart of either node would throw the protocol into an arbitrary state from which it could not recover of its own accord. Indeed, the initial correct operation of the scheme assumes that the value of the Sender's credit register is set to the size of the Receiver's queue and that the Receiver's queue is empty.

The imposition of local checking and correction techniques on such a protocol enables it to self-stabilise. Local checking would involve the Sender node ascertaining that a *local predicate* holds true at all times. A local predicate is a locally checkable condition which, if true, indicates that the link is behaving normally. In this case the predicate that must hold true is that the number of credits in the Sender's credit register added to the number of messages in the Receiver's queue is equal to the maximum credit register value. If this predicate is false then the system needs to be locally corrected. Local checking involves the Sender node issuing a request to query the state of the Receiver node; specifically, what number of messages is in its queue? The Receiver node responds with a reply message indicating its present state. If the predicate does not hold true, then the Sender node can send a message to the Receiver to correct the link and stabilise the protocol.

Varghese has shown that if a local checkable protocol is also *one-way checkable* then the protocol may use *local checking by periodic sending of state* instead of local snapshots if each local predicate is also *separable.* A separable local predicate is one that can be split into two one-way predicates for each direction in the link subsystem. If a protocol is one-way checkable, it is faster and cheaper and simpler than a protocol that uses snapshots and resets.

While the concept of local checking may be applied to many protocols, local correction is often not applicable due to an inability to impose such sufficient conditions on the underlying protocol for its successful self-stabilising operation. In cases where local correction is not possible, Varghese suggests that global correction mechanisms may be used. One such global correction mechanism is the distributed global reset protocol proposed by Arora *et al.* [Aro94]. While global resetting replaces the notion of local correction by local reset, it may still be triggered by a local checking mechanism. However, such a global reset algorithm is comprised of leader election, spanning tree construction and diffusing computation elements. The first two components clearly violate the assumptions that have been made about the flexible system. Leader election introduces hierarchy into the system and spanning tree construction is itself analogous to multi-hop route construction.

Composite systems may be designed around separate system components that address different system faults or unanticipated changes. If individual system components can be stabilised, then it is possible to stabilise a larger composite system. In the case of network protocols this results in partitioning the protocol into separate elements that are capable of correcting their own state. While the notion of self-stabilisation is not explicitly referenced in the published networking literature, the protocols that are components of networking

systems are generally self-stabilising. Stateless protocols such as IP are stabilising by definition while stateful protocols such as RIP, OSPF, AODV and OLSR are stabilising, whether by accident or design.

In summary, protocols that incorporate self-stabilising mechanisms are robust to the challenges of asynchronous, lossy, message-passing distributed systems. There are many approaches to the design of self-stabilising protocols, varying from global snapshots and distributed resets which require the centralised collation of system information followed by the network-wide resetting of the local system entities to the localised proposals put forward by Varghese which may be used in general networks subject to the imposition of some conditions. In terms of time complexity and messaging overhead, the most cost effective self-stabilising protocols are ones that make use of local checking and correction techniques. Appendix B provides a more detailed primer on the concept of self-stabilisation by local checking and local correction.

6.4.4 Summary

Both the concept of router address auto-configuration and the technique of self-stabilisation lend themselves to the task of designing an auto-configuration component. The literature on router address auto-configuration was analysed even though the configuration of unique addresses and the configuration of uniform choice of routing protocols have orthogonal configuration objectives; the former seeks nodal-configuration heterogeneity, the later seeks nodal-configuration homogeneity. Whilst acknowledging the differences between the two objectives, it was noted that addresses and routing protocols are both dependent and variable parameters of the network. Both addresses and routing protocols have non-local significance in a network as collaborating nodes must use a combination of unique addressing and uniform routing protocols to provide end-to-end connectivity. The authors of the published literature acknowledge the difficulties of designing a distributed and collaborative configuration mechanism.

The analysis of the literature on address auto-configuration revealed that there are two primary approaches to the collaborative auto-configuration of addresses in ad hoc networks. The published techniques use either consistent stateful approaches or possibly-inconsistent stateless approaches. The stateful techniques involve all nodes having complete knowledge of the address configuration of every node in the network. A node is configured with a new address based on the active cooperation of all other nodes. There is no ambiguity in the network as regards the configuration of any node. On the other hand, the stateless techniques adopt a more localised approach in which global state knowledge is not required; rather, nodes tentatively configure addresses based on local information and configuration conflicts are resolved as and when they are detected. The literature has shown the signalling overhead involved in the stateful techniques to be high, whilst the configuration inconsistencies incurred by the lower-cost stateless techniques are readily surmountable. The address configuration literature also draws attention to the use of various types of network partition identifiers as a means to enable mobile wireless ad hoc networks to control the effects of network partitioning and merging.

It was noted in the description of self-stabilisation that the main thrust of the concept is to allow a distributed system to converge, within a finite number of steps, from an arbitrary state to one that exhibits the desired system behaviour. To put self-stabilisation into the context of the proposed network-layer auto-configuration

component; an arbitrary state exists when a migrating node or merging network impinges on another network region which was heretofore exhibiting *desired system behaviour*, i.e. all nodes were using the same protocol. If the auto-configuration component is designed to be self-stabilising, it could correct the *arbitrary behaviour* of the migrating or merging nodes so that the network operates correctly within a bounded time. Self-stabilisation techniques operate by imposing verifiable conditions on the underlying protocol or algorithm which may be continually checked and corrected, if necessary, thus stabilising the system with respect to the desired system behaviour. Furthermore, the non-terminating nature of the self-stabilisation approach, i.e. that a self-stabilising protocol or algorithm constantly detects and corrects its behaviour, is suited to the mobile ad hoc networking environment as such an open, autonomous system undergoes continual change.

The self-stabilisation literature has examined a wide variety of self-stabilisation approaches for many different types of distributed systems. Many self-stabilisation approaches described in the literature propose the use of global detection and correction techniques, techniques which are not suited to mobile wireless ad hoc networks. However, the literature has identified the local correction and detection approach as being suitable for application in a distributed setting such as that exemplified by mobile wireless ad hoc networks. Such self-stabilisation techniques offer the most robust and low-cost solution to the design of a configuration component.

Indeed, when self-stabilisation techniques and address auto-configuration techniques are viewed in conjunction with one another, the parallels between the two general approaches of each technique become evident. Both the stateful address auto-configuration techniques and the global detection and correction self-stabilisation techniques are equally inefficient in the context of the distributed system, especially when the distributed system is that represented by a realistically modelled wireless mobile ad hoc network. The dynamic and open topology, the unbounded population size and the multi–hop nature of the communications involved in each approach introduce signalling and global-state requirements that may overwhelm, or at least place heavy demands upon, the primary network-layer components. As the routing protocol is one of the primary network-layer components, the overhead involved in configuring the choice of routing protocol must not impinge on the operation of the routing protocol itself, rather it must be sympathetic to the fundamental signalling characteristics of the routing protocol in use. On the other hand, both the stateless router address auto-configuration techniques and the local detection and correction techniques take a less benign view of the characteristics of distributed systems with regard to organising and effecting a distributed collaborative task.

It was also noted that composite systems, i.e. systems that are composed of multiple individual components, may be made self-stabilising if the individual components are themselves self-stabilising. Routing protocols such as AODV, DSR and OLSR are inherently self-stabilising. Therefore, other distributed components that are developed to work in tandem with these self-stabilising routing protocols, such as the auto-configuration component, should also adhere to a similar design objective so that the entire system has the desirable qualities associated with self-stabilising distributed systems.

6.5 The Auto-Configuration Component

The auto-configuration component incorporates the elements identified in the blueprint described in Section 6.2.3. The auto-configuration protocol is presented as two separate sub-protocols; the Stateless Configuration Initialisation (SCI) protocol and the Configuration Conflict Detection and Resolution (CCDR) protocol. Firstly, an overview of the component is presented in the next section. This overview sets out the design rationale for the component which is based on the foregoing discussions in this chapter. Then in the following section, the state variables and messages used by the protocols in this component are introduced and explained. Finally, in the last two sections, the SCI protocol and the CCDR protocol are presented.

6.5.1 Overview of Component

The overriding aim of the auto-configuration component is simple; each node should operate the same protocol as its neighbours; nodes should be enabled to dynamically auto-configure their network-layers with respect to their choice of routing protocol so that the state of the network, when viewed as the union of the states of its constituent nodes, is consistent and homogeneous regardless of the naturally occurring disruptive events that occur.

More general objectives for the design of this protocol mandate that it is *robust*, *extensible* and generates a *low overhead*. The component should be robust enough to deal with each of the naturally occurring network events that were described in Section 6.2, stabilising the network configuration in a timely manner. It should also be extensible so that other protocols may be accommodated in future systems. The manner in which such system upgrading or modification would take place is beyond the scope of this thesis, but some issues surrounding it are briefly discussed in Chapter 11. As previously stated, the auto-configuration component is a secondary network-layer component and it should not generate overhead which would conflict with the operation of the primary network-layer routing components, or which runs counter to the general approach to mobile wireless ad hoc network protocol design.

Bearing the attributes of both stateless router address auto-configuration and localised self-stabilisation in mind, it is proposed that the flexible multi-protocol network-layer should employ some aspects of these techniques that are suited to the network-layer routing protocol auto-configuration tasks. As a node only needs to be compatible with its immediate one-hop neighbours, this allows the protocol to be designed around a near-stateless approach which is similar to the stateless address configuration protocols that combine weak global consistency with a local configuration conflict detection algorithm. The scheme is near-stateless, rather than stateless or stateful, as, while it does not require complete knowledge of the global state of the network, it does rely on information about the state of the local neighbourhood. The need to record and evaluate the local state of the network introduces a minimal degree of statefulness into the protocol. The last section examined some of the issues surrounding a node's ability to perceive its local environment with regard to the configuration of its neighbouring nodes. The characterisation of the degree to which different types of nodes can appreciate their environment is fundamental to the design of the auto-configuration component.

It is also necessary to bear in mind that two other complementary flexibility components are proposed in this thesis; the internetworking component and the reconfiguration component. The reconfiguration component proactively goes about reconfiguring regions of the network in response to the underlying networking dynamics. The reconfiguration component sets the boundaries between regions of nodes where the underlying dynamics of the network warrant the use of different ad hoc routing protocols. The internetworking component facilitates this regional heterogeneity by enabling end-to-end connectivity across a heterogeneously configured network.

In order to enable the auto-configuration of nodes, the reconfiguration of regions of nodes, and the internetworking of heterogeneous regions, the flexible multi-protocol network-layer allows nodes to change from one protocol to another at runtime and to simultaneously operate two routing protocols within the same network-layer. A node which operates two ad hoc routing protocols simultaneously is referred to as a Boundary Node. Boundary Nodes have special significance in terms of regional internetworking, as will be discussed in Chapter 8.

Figure 6.11 provides a simple illustration of the possible runtime changes that a node can make to the configuration of the routing protocols operating in its network-layer. As illustrated in Figures 6.11a, a node can change from using AODV to using either of the other two available routing protocols if either the auto-configuration or reconfiguration component triggers such a change. Nodes can also simultaneously operate two routing protocols. Figure 6.11d depicts a node that cycles through a series of protocol configurations; (anticlockwise) from OLSR to OLSR and AODV simultaneously to AODV to AODV and DSR simultaneously to DSR and OLSR simultaneously and back to OLSR. This feature is triggered and controlled by the auto-configuration component. The specific detail of the dynamic changing of protocols and the simultaneous operation of protocols is discussed in Chapter 8.

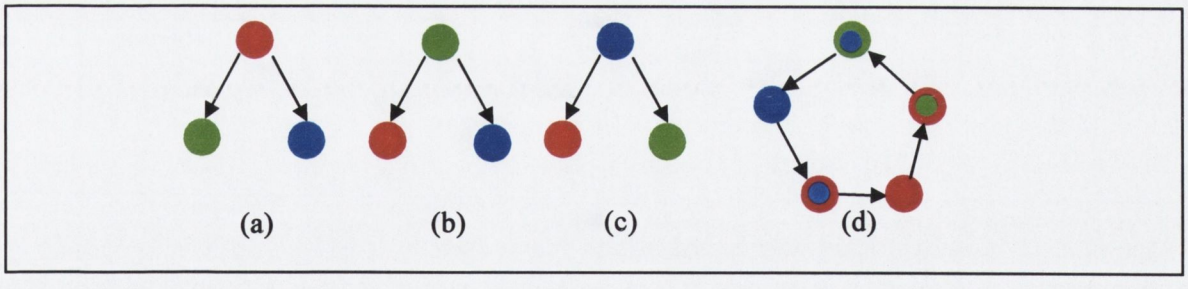


Figure 6-11 Nodes Can Change Protocols Dynamically and Operate Two Protocols Simultaneously

The auto-configuration protocol incorporates the self-stabilisation techniques of local checking and local correction to develop a robust solution suited to the harsh distributed computing environment exemplified by a wireless mobile ad hoc network. The protocol is completely distributed; each node takes local independent actions based on a combination of local state variables and information received from neighbouring nodes in packets that are issued by those nodes' primary network-layer components; indeed the absence of information from neighbours also informs a node's decisions.

The design of the auto-configuration protocol is described in the following sections. The auto-configuration component is split in to two distinct protocols; the Stateless Configuration Initialisation (SCI) Protocol and the more complex Configuration Conflict Detection and Resolution (CCDR) Protocol. The SCI protocol

deals simply with the configuration of a single node when it boots up. The node's routing protocol is tentatively configured and some state variables necessary for the operation of the CCDR protocol are configured. Once the SCI protocol has configured those network-layer parameters it terminates and the CCDR protocol becomes operable. The CCDR protocol is non-terminating; it operates as long as the node is alive.

The protocols are presented in a number of stages. Firstly, Section 6.5.2 provides a brief primer on the specific self-stabilisation techniques that will be used to design the CCDR protocol. Having identified the self-stabilisation techniques of local checking and correction as being suitable techniques to incorporate into the design of an auto-configuration protocol, it is necessary to thrash out some of the details of these two techniques so that their impact on the component design may be understood. The local variables and messages that form the basis of the protocols are discussed in Section 6.5.3. Finally, in Sections 6.5.4 and 6.5.5, the SCI protocol and the CCDR protocol are presented.

6.5.2 Local State Variables and Node Status Messages

To introduce the local state variables and messages that are used in the description and operation of the protocols, it is useful to consider Figure 6.12 which depicts a network that is composed of two contiguous regions of nodes. The region on the left is configured to use DSR and the region on the right is configured to use OLSR. Each node-to-node pair of connected nodes forms a link subsystem.

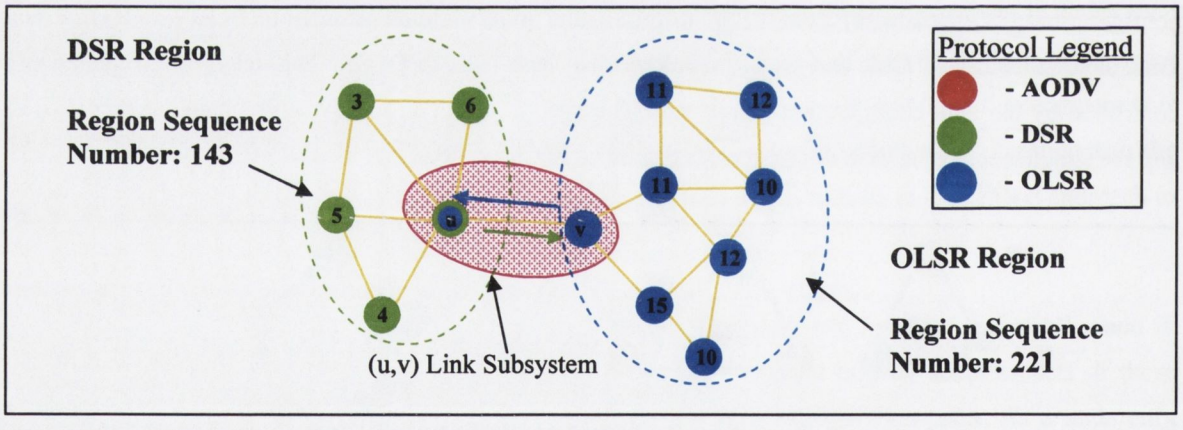
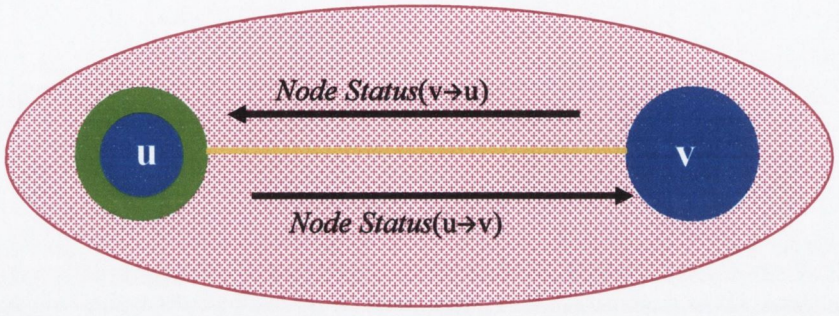


Figure 6-12 A Network, Two Regions and A Link Subsystem

The (*u*, *v*) link subsystem shown in Figure 6.12 is depicted in full detail in Figure 6.13; the state of the link subsystem is fully described by the local states of the two nodes and the messages that intervene between them. Each node stores a number of state variables that are used in the operation of the protocol, as depicted in Figure 6.13. The details of these state variables and the Node Status messages are described below:

Active Routing Protocol: This entry simply records the routing protocols that the node is configured to use. The first protocol to be configured in the node is referred to as the **A** protocol and the second one to be configured, if the need arises, is the **B** protocol. The protocols' Region Sequence Numbers are also tagged as A or B protocol entries. As shown in Figure 6.13, Node *u* records DSR as its A protocol and OLSR as its B protocol, while Node *v* records OLSR as its A protocol.

Region Sequence Number: The Region Sequence Number uniquely identifies a region of homogeneously configured nodes to which the node is presently connected. The Region Sequence Number is motivated and discussed in the next section. Node *u* records its Region Sequence Numbers as A: 143 (for the DSR region) and B: 221 (for the OLSR region) and Node *v* records its Region Sequence Number as A: 221.



Local State(u)

u: Active Protocol : A:**DSR** / B:**OLSR**
u: Region Sequence No. : A:**143** / B:**221**
u: Neighbourhood Table:

Neighbourhood Table				
Entry	**Neighbour ID: Protocol**	**Active Protocol**	**Region Sequence No.**	**ESV Data (Degree)**
0	4: A	A: DSR	A: 343	2
1	5: A	A: DSR	A: 343	3
2	3: A	A: DSR	A: 343	3
3	6: A	A: DSR	A: 343	2
4	v: B	B: OLSR	B: 701	3

Local State(v)

v: Active Protocol(s) : A:**OLSR**
v: Region Sequence No. : A: **221**
v: Neighbourhood Set :

Neighbourhood Table				
Entry	**Neighbour ID: Protocol**	**Active Protocol**	**Region Sequence No.**	**ESV Data (Degree)**
0	11: A	A: OLSR	A: 701	4
1	15: A	A: OLSR	A: 701	3
3	u: A	A: DSR B: OLSR	A: 343 B: 701	5

Figure 6-13 Local State of the (*u*, *v*) Link Subsystem

Node Status Messages: The link subsystem is also described by the information contained in what will generically be referred to as Node Status messages. A Node Status message is carried in *every* packet received from *every* neighbouring node, regardless of the main payload of the packet. All flexible network-layer nodes encapsulate their network-layer payload, whether it is a control packet or a data packet from a higher layer, within a Network-Layer Packet. The Network-Layer Packet was described in Chapter 4. A Node Status extension is appended to the all Network-Layer headers. As the network-layer detects duplicated and out-of-order messages using the unique combination of each packet's unique Source Address field and unique Packet Identifier field, each Node Status message is only processed once by the receiving node's auto-configuration component. The Node Status message has local scope only and is not repeated by any node.

The Node Status Extension is illustrated in Figure 6.14. It contains the Region Sequence Number and the External State Vector (ESV) data pertaining to the node that issued the packet. The External State Vector, which was described in the last chapter, details the Mobility, Degree and Traffic characteristics of the node that issues the Node Status message. Only the Degree value in the ESV field is used in this component as it

indicates whether the issuing node is connected or not. A connected node will declare a non-zero Degree value, whereas an isolated node will declare a Degree value of zero.

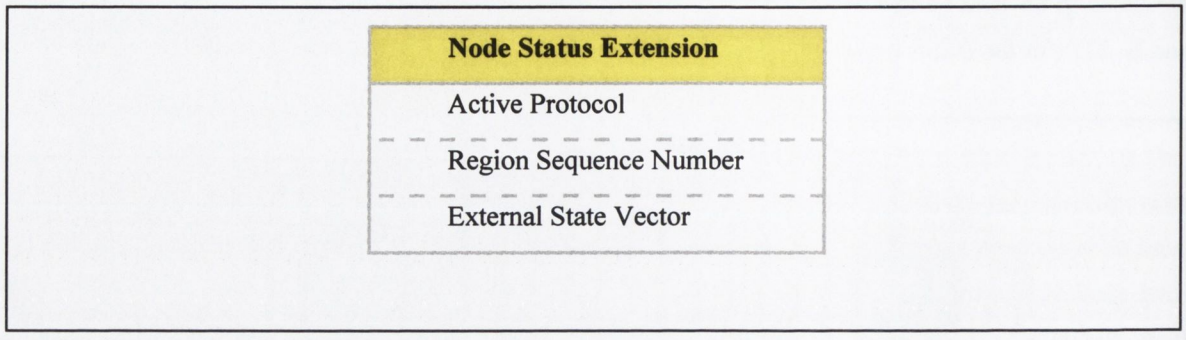


Figure 6-14 Node Status Message Extension

If the node issuing the Node Status messages is operating two protocols simultaneously, this means that it is currently connected to two differently configured networks. As such, the node will maintain the Region Sequence Numbers of the two regions to which it is connected. The node will also collate information about the networking conditions pertaining to those two regions. Such a node will attach two Node Status headers to *every* packet that it issues. Thus, the complete state of the node's protocol configuration, i.e. that it is simultaneously operating two protocols, is revealed to nodes receiving the Node Status messages. Node *u*, in Figure 6.12, would attach two Node Status headers to the Network-Layer header of each packet that it broadcasts, one for the DSR region and one for the OLSR region.

As noted in the Section 6.4.3, one-way checking is sometimes referred to as 'periodic checking by sending of state'. The periodicity of the sending of such state information determines how quickly a fault is detected. It has been observed that the three candidate protocols have varying signalling characteristics, ranging from the frequency-consistent signalling of OLSR to the aperiodicity of the signalling generated by the reactive routing protocols; the signalling of reactive protocols is related to the traffic demands on the network. The Node Status packets, which carry the state information that is used to do the periodic one-way checking of the local predicates, are *always* carried in the headers of the packets generated by the primary network-layer routing components, i.e. the routing protocol components and the packet-switching components. It will be shown in Section 6.5.4 that the variable nature of the periodicity with which these packets are sent does not diminish the capability of the auto-configuration protocol.

Extended Neighbourhood Table: The basic Neighbourhood Table was described in Chapter 4. Each entry in the Neighbourhood Table is managed in accordance with the associated routing protocol, i.e. each neighbour is entered and removed as dictated by the implicit or explicit route deletion rules for each ad hoc routing protocol. Two examples of the Extended Neighbourhood Table are illustrated in Figure 6.13. As shown in Figure 6.13, entries 0, 1, 2 and 3 in Node *u*'s Neighbourhood Table are associated with its A protocol, DSR, and entry 4 is associated with its B protocol, OLSR. The protocol identified in the *Neighbour ID: Protocol* column indicates which one of the Node *u*'s Active Protocols detected, and manages, the entry. The *Active Protocol* column indicates which protocols are active in the neighbouring node. The Extended Neighbourhood Table contains additional data fields for each neighbour entry. The modified table contains three extra data fields to record the *Region Sequence Number* and *ESV* information that is carried in the Node

Status packets which are issued by each neighbouring node. As noted earlier, only the Region Sequence Number and the ESV Degree values are used in the operation of the auto-configuration protocol. The receipt of two Node Status messages in a single packet from the same node indicates that the sending node is simultaneously operating two ad hoc routing protocols. Referring to entry 3 in Node *v*'s Neighbourhood Table, it is possible to see that the neighbour entry for Node *u* reflects the fact that it is operating DSR and OLSR simultaneously.

6.5.3 Stateless Configuration Initialisation (SCI) Protocol, Region Sequence Number

■ **Introduction**

The auto-configuration component is triggered as soon as a node is booted up; the component immediately engages the Stateless Configuration Initialisation protocol. As part of the initialisation process, each node must configure its Region Sequence Number in addition to its ad hoc routing protocol. The description of the Region Sequence Number and the Stateless Configuration Initialisation protocol are presented in tandem as they are dependent on each other.

The Region Sequence Number gives a unique identity to regions of nodes that are not like-configured. As illustrated in Figure 6.13, the DSR region has an identity of 143 and the OLSR region has an identity of 221. The Region Sequence Number is constructed such that these two regions cannot have the same identity while heterogeneously configured.

As the Configuration Conflict Detection and Resolution protocol uses the self-stabilising techniques of local checking and local correction it is necessary to impose a partial order on the link subsystems so that they may be corrected locally. Otherwise, since the link subsystems are not independent, the correction (or reconfiguration) of one link subsystem may cause another link subsystem to become invalid. The reliance of the local correction techniques on the existence of dependency-stable local predicates is detailed in Appendix B. However, within the context of this work, groups of link subsystems operate the same ad hoc routing protocol on a regional basis. Local correction will be applied locally by nodes with reference to their regional configuration. In short, a partial order has to be imposed on regions which are not like-configured. Applying the same reasoning used for local correction of a link subsystem to a regional subsystem of the network, i.e. applying it to groups of link subsystems, it is possible to construct stable local predicates if an acyclic dependency can be enforced on heterogeneous regional subsystems of the network. The use of such a unique region identifier for link subsystem correction is further described in Section 6.5.4.

■ **Choosing an Appropriate Region Identifier**

The purpose of a region identifier is to give each homogeneously configured region of nodes an identity that allows associated nodes to recognise each other and which may also be used to create an artificial hierarchy between heterogeneously configured regions when configuration conflicts occur. Such an identifier must be unique to each homogeneous region of nodes and must also be independently computable. The identifier

must also be independent of the actual routing protocol configuration of that region and it must be partition-safe.

The identifier must be unique for the simple reason that it is to be used as an arbitrary tie-breaker between regions of nodes. If the identities of two regions are guaranteed to be unique then it is possible to use the regions' identities to arbitrarily influence their actions in a resolution of some configuration conflict. The identifier should be independently computable in order to reduce the overhead necessary to maintain it. The address auto-configuration scheme described in [Sun03b] used a centralised approach that required a hierarchical authority node to initialise and maintain the network identifier in addition to periodically broadcasting it to every other node in the partition. Such an approach is not viable in the networking system proposed in this thesis under the design constraints that have been enumerated in Chapter 4.

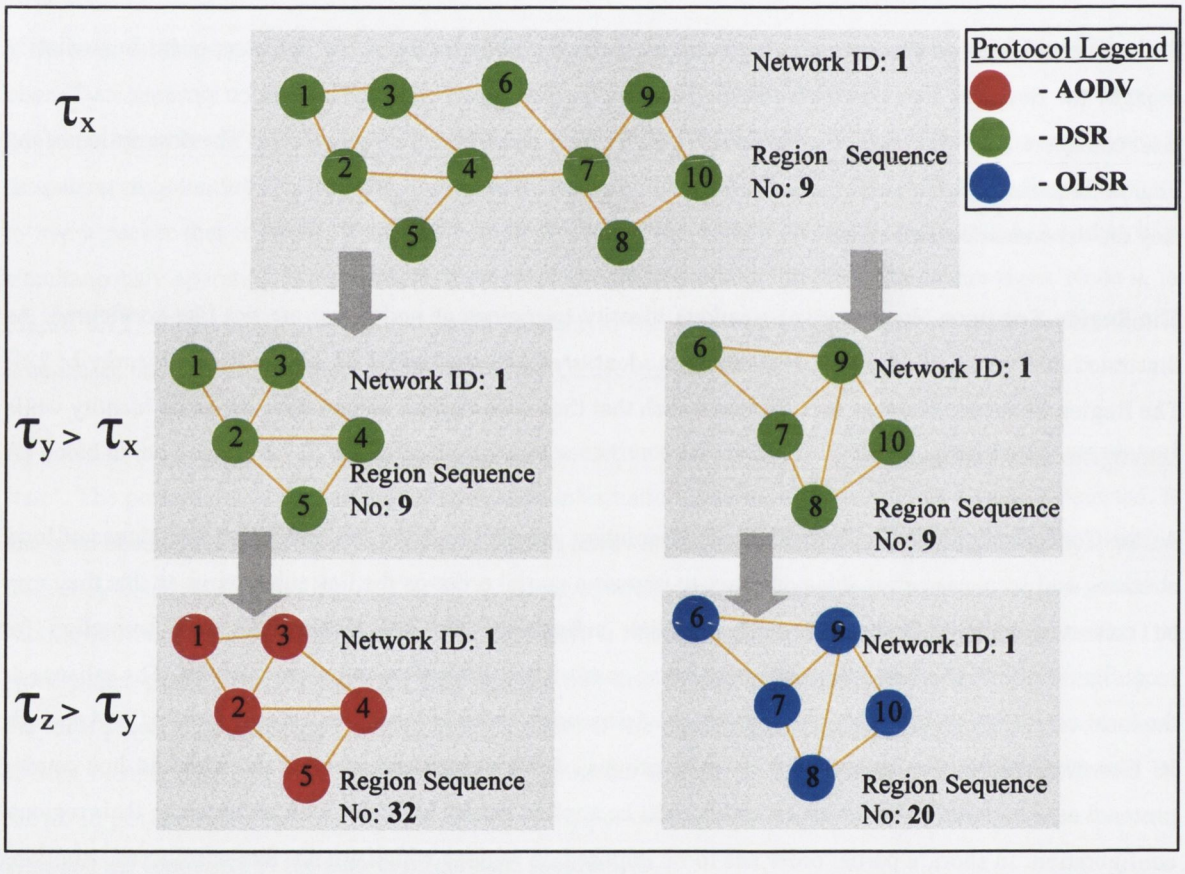


Figure 6-15 Network Identifiers Must be Partition-Safe

The identifier must be independent of the actual routing protocol configuration for the region, i.e. the value of identity should not be determined solely by the node's *current* routing protocol. The auto-configuration component treats all of the available routing protocols equally and does not distinguish between their capabilities, whereas the reconfiguration component actively reconfigures nodes based on its knowledge of the perceived networking conditions and of the routing protocols' different performance profiles. Consequently, the identity should not favour any particular protocol. Basing an identity on the current protocol's identity would introduce an arbitrary, yet permanent advantage or disadvantage to some protocol. It was noted in Chapter 4 that the three candidate protocols use unique prime numbers to identify them in packet headers; DSR: 2, AODV: 3 and OLSR: 5. If a region of DSR nodes and a region of OLSR nodes were

to be identified by their protocol values, then the OLSR region would always have a higher region identity than a DSR region resulting in a permanent, yet artificial, regional hierarchy.

The identifier is also required to retain its uniqueness when faced with a partition event. The motivation for demanding this property is explained by way of an example. Figure 6.15 depicts a network at time τ_x, which consists of ten nodes configured to use DSR and identified by the same network identifier, 1. By time τ_y, the network has been partitioned and two network regions now exist. The two network partitions are still configured to use DSR and are still using the same original network identifier, 1. If these two partitions were to merge again, while still identically configured, then there would be no configuration conflict and no need to rely on the region identifier. However, by time τ_z the two partitions have reconfigured themselves to use alternative protocols as the networking conditions have changed. By time τ_z, one partition has been configured to use AODV while the other is configured to use OLSR. However, the two partitions are still using the same original network identifier, 1, even though the two networks are now heterogeneously configured. If the two partitions were to merge at this stage, there would be a configuration conflict but the regions would still share the same identity.

The stateful MANETconf [Nes02] protocol uses the lowest node address in a given network partition as the partition identifier. The uniqueness and partition-safeness of this identity marker is assured as the protocol guarantees the node addresses to be unique at all times. Since total state information is held at each node in the network partition, each node can easily ascertain the lowest address in the partition, and thus each node may ascertain the network partition identity. A similar approach was taken in [Bol02] which used the 2-tuple of the highest partition address and the address to size to identify the network partition. This type of stateful information is not available to nodes in the proposed networking system.

■ **The Region Sequence Number**

In order to address each of the abovementioned requirements, the proposed network identifier takes the form of a novel sequence number that cycles though a sequence of prime numbers dictated by the protocol-to-protocol transitions that occur within each network partition. This number is referred to as the Region Sequence Number.

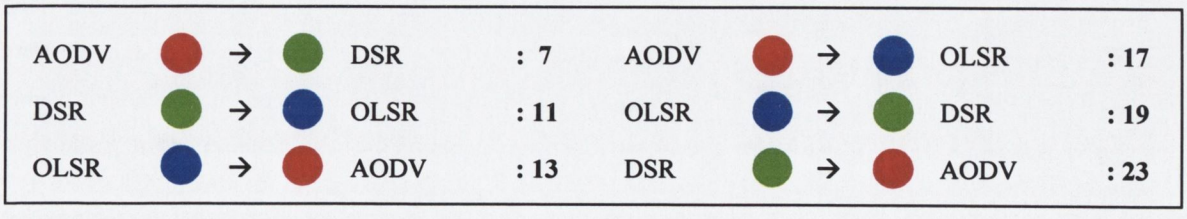


Figure 6-16 Unique Protocol-to-Protocol Transition Identifiers

Each unique protocol-to-protocol transition, i.e. a change from using one protocol to using another protocol at runtime, is assigned an arbitrary prime number value. The values for each of the six protocol-to-protocol transitions possible in this networking system are listed in Figure 6.16. The Region Sequence Number is independently incremented by a node whenever it undergoes a protocol change, e.g. if a DSR-configured

node changes over to OLSR it increments its Region Sequence Number by 11. If a region of nodes undergoes such a change, then each node can independently modify its Region Sequence Number so that the entire region then has the same identity.

Referring to Figure 6.17, it is possible to see that a node's identifier, which will be seeded by one of the three unique protocol identifiers, will then cycle through a sequence of prime numbers. As the sequence number is dependent on the protocol-to-protocol transitions, and as the transitions are identified by these unique prime numbers, then two regions can only ever have the same Region Sequence Number if they were initially configured to the same protocol and then subsequently experienced the same protocol-to-protocol transitions.

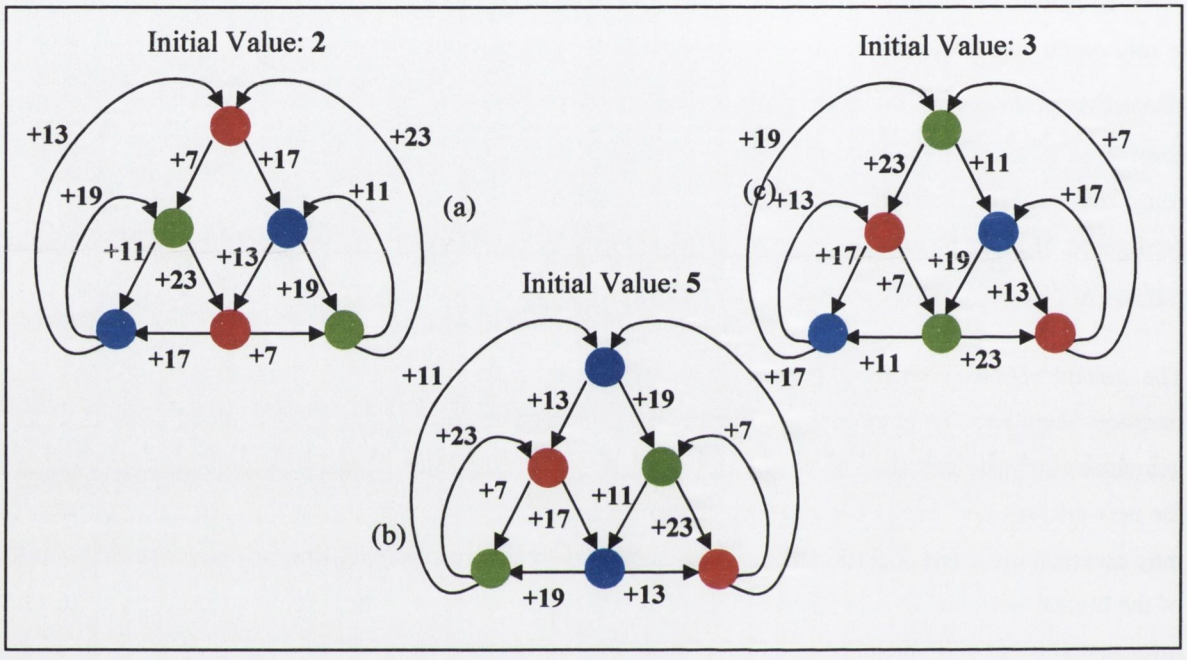


Figure 6-17 The Three Possible Region Sequence Number Cycles

In sum, two regions can never have the same Region Sequence Number unless they are like-configured and, importantly, the Region Sequence Number is independent of the current regional routing protocol configuration.

■ Stateless Configuration Initialisation

The unconfigured node tentatively configures itself by executing the following steps. Firstly, the node randomly chooses one of the routing protocols available to it; AODV, DSR or OLSR. As illustrated in Figure 6.18, the node then increments its Region Sequence Number from zero to the protocol's prime number value, e.g. a node that picks DSR will now have a Region Sequence Number of 3. Then, in the second step, the node randomly chooses one of the protocol-to-protocol transition steps that it can take.

Again referring to Figure 6.18, a node that initially chose DSR can now randomly choose between AODV and OLSR. The node then increments its Region Sequence Number by the protocol-to-protocol transition value. For example, if the node had initially chosen OLSR and then transitioned to AODV, it would now

have a Region Sequence Number of 18. The node now has its A protocol configured and that completes the costless stateless initialisation process.

The two-round process has the effect of giving each protocol a chance to dominate in the early stages of network creation as each region may be initialised with a high or low Region Sequence Number. As illustrated in Figure 6.18, a DSR node may have a Region Sequence Number of either 9 or 24 and an OLSR node may have a Region Sequence Number of either 14 or 19. This two-round randomised initialisation process gives each protocol a chance to lock-in other nodes in the early stages of a network's life-cycle.

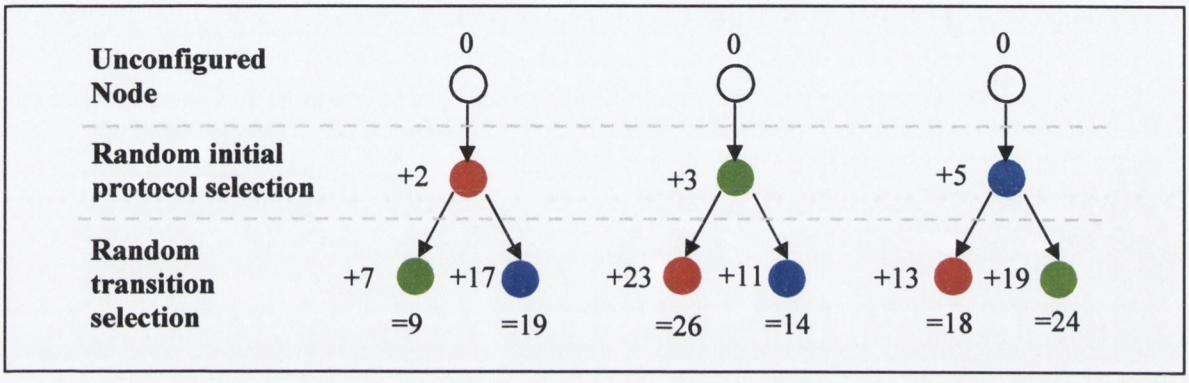


Figure 6-18 Two-Round Initial Random Protocol Selection

The Region Sequence Number is 1 byte long, i.e. it ranges from 9 to 255, and uses the following wrap-around-*proof* technique to compare Region Sequence Numbers. This technique is used in the OLSR protocol [Cla02]. The Region Sequence Number RSN1 is said to be "greater than" the Region Sequence Number RSN2 if, and only if: (RSN1 > RSN2 **&&** (RSN1 – RSN2) <= (255-9)/2) || (RSN2 > RSN1) **&&** (RSN2 - RSN1) > (255-9)/2). Thus when comparing two Region Sequence Numbers, it is possible, even in the presence of wrap around, to determine the order of the sequence numbers. Also, when a Region Sequence Number cycles through, or to, zero, the protocol-to-protocol transition that takes it past 255 is matched to the nearest initialisation value for that protocol, e.g. if a DSR to OLSR transition wraps around to 12, the number is incremented to 14 so that it aligns with nodes which may have just initialised to OLSR. The lowest possible initial value is 9. This process preserves the integrity of the Region Sequence Number.

The configured but isolated node is now in a legal state and is ready to connect to and network with other nodes. However, other nodes may not be like-configured and so the node engages the Configuration Conflict Detection and Resolution protocol.

6.5.4 Configuration Conflict Detection and Resolution (CCDR) Protocol

■ Introduction

The second part of the auto-configuration component is built around a protocol that detects and resolves conflicting ad hoc routing protocol configurations in a fast, robust and low cost manner; a conflict occurs when a node-to-node link subsystem does not conform to the desired network state. The CCDR protocol uses closed, separable and dependency stable local predicates such that two one-way predicates may be imposed on each link subsystem in the network. The use of closed, separable and dependency stable local predicates

means that the cheap, fast and robust techniques of local one-way checking and local one-way correction may be used. The local predicate for the CCDR protocol will be described in terms of *isolated* and *connected* nodes, which have already been configured by the SCI protocol, and the Node Status messages that pass between the two nodes in each link subsystem. The CCDR protocol handles the type of naturally occurring disruptive events that were described in Section 6.2.2; *group initialisation*, *node join*, *migration* and *mergers*. These events are disruptive if they bring heterogeneously configured nodes within transmission range of each other; while the nodes can communicate on a node-to-node basis they cannot communicate on a multi-hop basis if they are incompatibly configured.

The use of local checking and local correction techniques within the CCDR protocol allows it to make use of the signalling generated by the existing primary network-layer components; the protocol does not need to originate any packets itself. Crucially, for a wireless mobile ad hoc network, the protocol also allows nodes to act autonomously and asynchronously to bring about a stable network configuration that conforms to the desired objectives.

■ Legal Global State

In the main, the legal global state only permits homogeneously configured node-to-node links. However, region-to-region links must also be permitted within this framework; two heterogeneously configured regions may be linked by means of intermediary nodes operating two protocols simultaneously to enable internetworking between the two regions. As regional variations in the configuration of nodes are an integral part of the networking system made possible by the flexible multi-protocol network-layer, the auto-configuration component must allow for the existence of intermediary nodes that enable regions to internetwork and co-exist. Isolated, or unconnected, nodes are also accommodated in the system; as such nodes are not connected to any other nodes they cannot conflict with any other nodes. Such isolated nodes are independent systems; their state is inherently valid so long as they remain isolated.

■ Local Predicates

In order to develop the CCDR protocol it is necessary to identify local predicates for the global system that are closed, stable and separable. It should be remembered that a predicate is simply an attribute or condition that can be checked or verified by some means. The auto-configuration link predicates will hold true when the local predicates at each of the link subsystems in the network hold true.

A local predicate of the network for the (u, v) link subsystem is based on a subset of the states of the (u, v) link subsystem. A local predicate is defined such that only when the local link subsystem of the network, on which it is defined, is in a valid state should the local predicate be satisfied or hold true. The link subsystem variables for the flexible multi-protocol network-layer node were described in Section 6.5.2. They numbered the node's active routing protocols, its Region Sequence Numbers and, if connected to other nodes, its Neighbourhood Table entries.

The key literature on self-stabilisation by local checking and local correction [Awe91], [Var92], did not allude to the effects of dynamic network topologies on the use of such techniques. However, it does show

that a self-stabilising system can be composed of individually self-stabilising components. Since the underlying ad hoc routing protocols are inherently self-stabilising, the effects of the changing topology of a wireless, mobile ad hoc network on the CCDR protocol can be mitigated by incorporating the information regarding such topology changes into the CCDR protocol's design.

The CCDR protocol is based on two separable local predicates. The first predicate checks that a node is configured correctly with regard to the last neighbour from which it has received a Node Status message. The second predicate ensures that a node operating two protocols simultaneously is configured correctly with regard to those nodes recorded in its Neighbourhood Table.

Predicate One: The first predicate states that Node *u*, the receiving node should operate at least one ad hoc routing protocol which Node *v*, the sending node, operates. It also says that whenever Node *u* and Node *v* do operate the same ad hoc routing protocol(s), that Node *u*'s Region Sequence Number should be greater than, or equal to, Node *v*'s Region Sequence Number for the corresponding ad hoc routing protocol.

Predicate Two: The second predicate is based on Node *u*'s Neighbourhood Table entries. It says that Node *u* should only continue to operate two ad hoc routing protocols if, for both protocols, it has at least one neighbour that is only operating each of those ad hoc routing protocols. For example, if Node *u* is operating both OLSR and AODV, then there should be at least one neighbour that *only* operates AODV and at least one neighbour that *only* operates OLSR.

Both of these predicates are closed; the second predicate is only dependent on local state variables which are controlled by the CCDR protocol itself or the underlying ad hoc routing protocol which adds and removes entries to and from the Neighbourhood Table. Likewise, the first predicate is only dependent on information carried in Node Status messages and the local state variables. The integrity of the Node Status message is guaranteed to be in sequence and unduplicated.

All corrective action taken by Node *u* is taken locally. Node *u* reacts to the violation of either predicate by changing its own configuration, either its ad hoc routing protocol or its Region Sequence Number, or both. The protocol operates by setting Node *u*'s values to correct the link subsystem rather than attempting to adjust the state of both nodes in the subsystem.

Predicate One: Local Checking and Local Correction. When Node *u* detects that it is not operating the same protocol as Node *v*, it must compare its position in the dependency hierarchy, relative to Node *v*'s reported position, before taking corrective action. Two dependencies are imposed on link subsystems in this protocol. Firstly, all isolated nodes are equally independent, i.e. each isolated node is a system consisting of one node, and they always yield to *any* other node, whether or not it is isolated or connected. As an isolated node lacks a connection to any other node, there is no reason to attempt to preserve its current configuration if it conflicts with another node with which it could network if alternatively configured.

An isolated Node *u* yields to Node *v* by taking on its configuration, i.e. adopting Node *v*'s ad hoc routing protocol setting and changing its Region Sequence Number to match Node *v*'s.

A connected Node *u*, i.e. a node that is part of a region consisting of at least two nodes, takes corrective action only if its Region Sequence Number is less than the Region Sequence Number of Node *v*. If Node *u* is operating the same protocol(s) as Node *v*, but Node *v* has a higher Region Sequence Number associated with its protocol(s) then Node *u* takes local corrective action by updating its lagging Region Sequence Number to match Node *v*'s superior declared Region Sequence Number. This corrective action ensures that a region of nodes that share the same protocol will eventually share the same Region Sequence Number, enabling each of the nodes to take the same unilateral, but regional, action with regard to that ad hoc routing protocol.

If Node *u* is only operating its A protocol and it receives a Node Status message from Node *v* indicating that Node *v* is operating one (or two) different ad hoc routing protocol(s), Node *u* yields to one of Node *v*'s ad hoc routing protocols if it has a lower Region Sequence Number by activating its B protocol and setting it to match the configuration of the corresponding protocol in Node *v*.

However, as the CCDR protocol's stability is built around the imposition of partial orders in the networking system, which is brought about by the use of the Region Sequence Number and the distinction between isolated and connected nodes, a situation can arise in which an isolated node within range of a connected node may continue to remain incomapatibly configured due to the system of partial orders used. This protocol penalty is depicted in the scenario illustrated in Figure 6.26(g).

Predicate Two: Local Checking and Local Correction. Node *u* receives interrupts from the Neighbourhood Table when a neighbour entry is modified, e.g. a neighbour entry is removed or a neighbour that was operating a single protocol starts simultaneously operating two protocols. When Node *u* is simultaneously operating two protocols itself, it ensures that at least one of its neighbours is operating one of Node *u*'s two protocols in single operation mode. If Node *u* does not have at least one such neighbour, it deactivates the first protocol for which it detects this change; the only reason for Node *u* to operate two protocols simultaneously is to facilitate internetworking between heterogeneous regions.

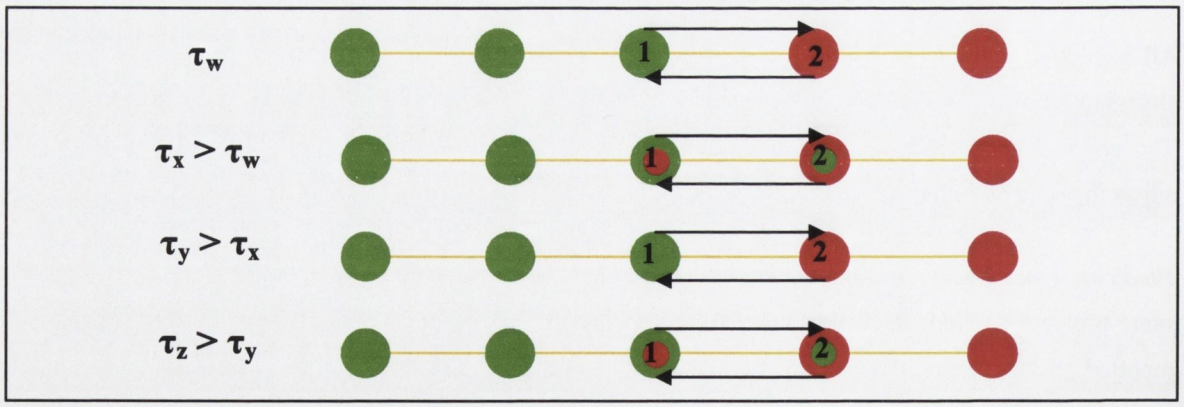


Figure 6-19 Region Sequence Numbers avoid Corrective Thrashing

The usefulness of the Region Sequence Number may be demonstrated by considering Figure 6.19, which depicts a simple chain of nodes; three are using DSR and two are using AODV. In the absence of an imposed partial order on the system, a node may attempt to correct the link subsystem *whenever* it detects a configuration conflict. For example, at time τ_w, Node 1 and Node 2 send each other Node Status messages.

Reacting to these messages, both nodes activate their second protocol so that they may internetwork. As there is no partial order between the DSR nodes and the AODV nodes, there is no restraint on both Nodes choosing to correct the link subsystem. However, according to Predicate Two, neither Node 1 nor Node 2 should not be operating in dual-mode as neither of them have a conflicting single-mode node as a neighbour at time τ_x. By time τ_y, both nodes have deactivated their second protocol and have reverted to single-mode operation. However, they are now back at the point where they discovered the initial conflict. It may be seen that without the imposition of the partial order on the system, the nodes could correctively thrash for ever, continually alternating between single and dual-mode operation without reaching a stable configuration.

In making use of the signalling generated by the primary network-layer components, the CCDR protocol is not slow to react to configuration conflicts. By its very nature, a conflict can only occur *when* a node attempts to contact another node. Both isolated nodes and connected nodes detect conflicts in a timely manner. Connected nodes, by definition, must signal regularly or else they lose contact with their neighbours. The ambient signalling provided by connected regions of nodes allows both isolated and connected nodes to detect and resolve any configuration conflicts that may exist. An isolated node in the presence of other isolated nodes can only detect and resolve conflicts if it starts signalling to those nodes and forces them to adopt its configuration. By definition, a node will start signalling if it wants to form a network with other nodes as a result of demands from higher layers. Such an isolated node would, by definition, issue *route requests* or *hello* packets, regardless of the type of ad hoc routing protocol. Such signalling will illicit *route responses* or *hello* packets from the newly connected and configured nodes, which were formerly quiet, isolated and possibly incompatibly configured, thus allowing normal ad hoc networking to occur.

■ CCDR Protocol Pseudo-Code

Each node sends a Node Status message for each protocol that it operates in the header of every packet that it broadcasts. Referring to Figure 6.20, it may be seen that Node *u* issues two Node Status messages. The first one, corresponding to its A protocol, indicates the A protocol as being DSR (*u*.Protocol.**A** = **DSR**), the Region Sequence Number associated with the A protocol as being 143 (*u*.RegionSeqNo.**A** = **143**) and the Degree of the node, as perceived by the A protocol's neighbour detection techniques, as being 4 (*u*.Degree.**A** = **4**). The second message indicates the *u*.Protocol.**B**, *u*.RegionSeqNo.**B** and *u*.Degree.**B** values. The receiving node parses these values in accordance with the CCDR protocol.

Pseudo-code for the CCDR protocol is presented in Figures 6.21, 6.22 and 6.23. Lines 1 to 42 show the local checking and local correction that is triggered by receipt of a Node Status message. Lines 43 to 51 show the local checking and local correction that is triggered by the self-stabilising ad hoc routing protocols making modifications to the Neighbourhood Table. A series of examples that illustrate the local checking and local correction process is then presented.

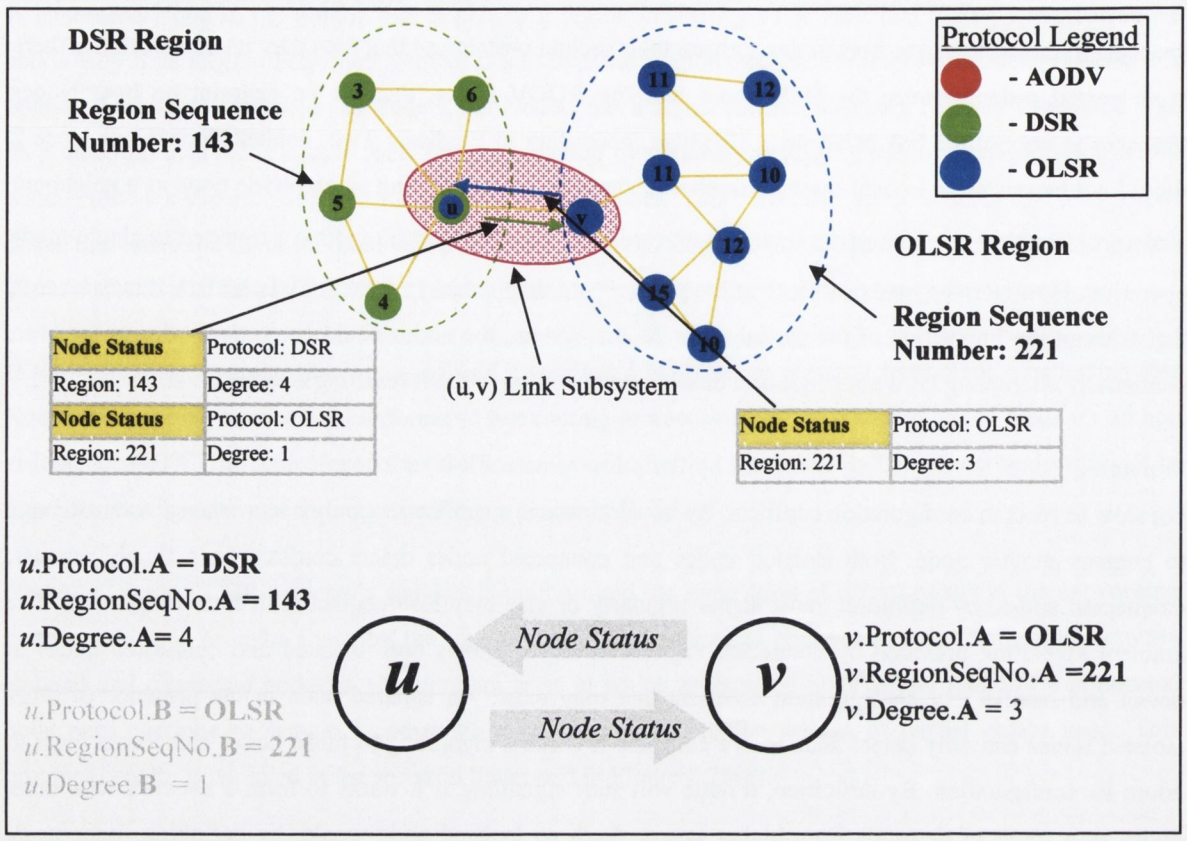
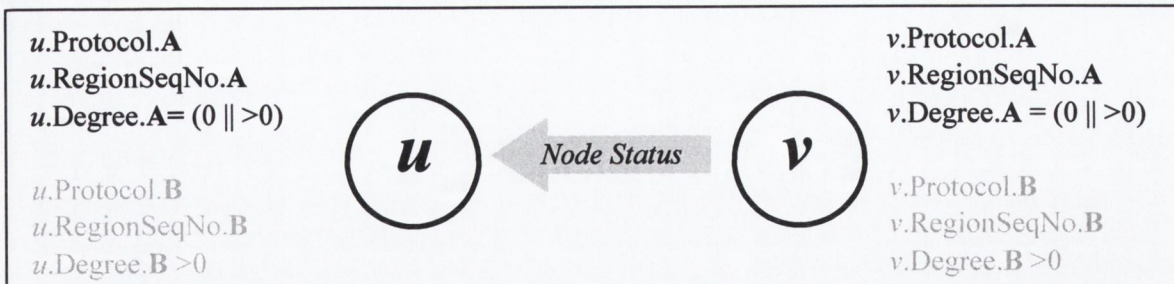


Figure 6-20 One-Way Checking: Interpretation of Node Status Messages



Node Status message received from Node *v*

```
1    if    (u.Degree.A == 0)      //Node u is isolated
           {
2    check if    (u. ActiveProtocol.A   ==   v. ActiveProtocol.A)
                 {
3    check       if    (u. RegionSeqNo.A    >=   v. RegionSeqNo.A)
                       {
4                            PREDICATE HOLDS
                       }
5    correct     else  {UPDATE u. RegionSeqNo.A= v.RegionSeqNo.A}
                 }
           else  {
6    correct     CHANGE     u.ActiveProtocol.A     =   v. ActiveProtocol.A
7                           u.RegionSeqNo.A        =   v. RegionSeqNo.A
8                           (u.Degree.A is updated by the ad hoc routing protocol)
                 }
9    else if (u.Degree.A > 0)       //Node u is connected
           {
10   check if    (u. ActiveProtocol.A   ==   v. ActiveProtocol.A)
                 {
11   check       if    (u. RegionSeqNo.A    >=   v. RegionSeqNo.A)
                       {
12                           PREDICATE HOLDS
                       }
13   check       else if (v.Degree.A   >   0)  //Node v is connected
                       {
14   correct             UPDATE      u. RegionSeqNo.A = v. RegionSeqNo.A
                       }
                 }
15   check  else if (v. ActiveProtocol.B is active)  //Node v is operating two
                 {                                    //protocols simultaneously
16   check       if    (u. ActiveProtocol.A   ==   v. ActiveProtocol.B)
                       {
17   check             if    (u. RegionSeqNo.A   >=   v. RegionSeqNo.B)
                             {
18                                 PREDICATE HOLDS
                             }
19   correct           else  {UPDATE u. RegionSeqNo.A=v.RegionSeqNo.B}
                             }
                       }
```

Figure 6-21 Conflict Detection and Resolution Protocol Pseudo-Code, Predicate One: Part I of II

```
20  check  if      (u.ActiveProtocol.B is active)   //Node u is operating two
                   {                                //protocols simultaneously
21  check          if      (u. ActiveProtocol.B  ==   v. ActiveProtocol.A)
                           {
22  check                  if      (u. RegionSeqNo.B   >=   v. RegionSeqNo.A)
                                   {
23                                         PREDICATE HOLDS
                                   }
24  check                  else if (v.Degree.A   >   0)
                                   {
25  correct                        UPDATE u. RegionSeqNo.B = v. RegionSeqNo.A
                                   }
                           }
26  check          else if (v. ActiveProtocol.B is active)   //Node v is operating two
                           {                                 //protocols simultaneously
27  check                  if      (u. ActiveProtocol.B     ==   v. ActiveProtocol.B)
                                   {
28  check                          if      (u. RegionSeqNo.B > =v. RegionSeqNo.B)
                                           {
29  correct                        UPDATE u. RegionSeqNo.B = v. RegionSeqNo.B
                                           }
                                   }
                           }
30  check  else if (u. ActiveProtocol.B is not active)       //Node u is only operating
                   {                                        //a single protocol
31  check          if      (v. Degree.A > 0)                 //Node v is connected
                           {
32  check                  if      (u. RegionSeqNo.A   <   v. RegionSeqNo.A)
                                   {
33  correct                ACTIVATE   u.ActiveProtocol.B = v.ActiveProtocol.A
34                                    u. RegionSeqNo.B  = v. RegionSeqNo.A
35                                    (u.Degree.B is updated by the ad hoc routing
                                      protocol)
                                   }
36  check                  else if (v. ActiveProtocol.B is active)
                                   {                        //Node v is operating two
                                                            // protocols simultaneously
37                                 if      (u.RegionSeqNo.A<v.RegionSeqNo.B)
                                           {
38  correct                ACTIVATE   u.ActiveProtocol.B = v.ActiveProtocol.B
39                                    u. RegionSeqNo.B  = v. RegionSeqNo.B
40                                    (u.Degree.B is updated by the ad hoc routing
                                      protocol)
                                           }
                                   }
41                         else    {NO CORRECTION–DEPENDENCY BARRED}
                           }
42                 } else { PREDICATE HOLDS }
           }
```

Figure 6-22 Conflict Detection and Resolution Protocol Pseudo-Code, Predicate One: Part II of II

```
Interrupt received from Neighbourhood Table: NEIGHBOUR ENTRY MODIFIED

43    if      (u. ActiveProtocol.B is active)  //Node u is operating two
              {                                // protocols simultaneously
54    check   if      (Node u's last B SINGLE MODE neighbour was removed)
              {       //May occur if last single mode node goes dual or leaves
45    correct         DEACTIVATE        u.ActiveProtocol.B
              }
46    check   else if (Node u's last A SINGLE MODE neighbour was removed)
              {       //May occur if last single mode node goes dual or leaves
47    correct         DEACTIVATE        u.ActiveProtocol.A
48                    REASSIGN          u.ActiveProtocol.A =  u.ActiveProtocol.B
49                                      u.RegionSeqNo.A    =  u.RegionSeqNo.B
50                                      u.Degree.A         = u.Degree.B
              // Node u is now only operating the A protocol
              }
51            else    {       PREDICATE HOLDS         }

              }
```

Figure 6-23 Conflict Detection and Resolution Protocol Pseudo-Code, Predicate Two

■ **CCDR Examples**

It may be useful to consider the following examples of link subsystems which apply the CCDR protocol in varying node-to-node scenarios. CCDR examples (a) to (i), below, examine a series of possible link subsystems that may occur within a network consisting of nodes operating the flexible multi-protocol network-layer and which have access to AODV, DSR and OLSR. For each example, the state of the two nodes making up the link subsystem, Node *u* and Node *v*, are described in terms of the pertinent state variables; Active Protocols, Region Sequence Numbers and Degrees. If a correction of the link subsystem has been necessary, the corrected local state is also shown. For the sake of brevity, only example (d), Figure 6.25 (d), is explained as an aide to understanding the format of the examples. This link subsystem consists of Node *u* operating OLSR and Node *v* operating DSR. Both nodes are connected, as indicated by their non-zero Degree values. When Node *u* receives Node *v*'s Node Status message it parses it according to the CCDR protocol; specifically at lines 30-35, since Node *u* is only operating a single protocol and Node *v*'s Region Sequence Number is higher. Node *u* corrects the link subsystem by activating its B protocol and setting it to the values of Node *v*'s configuration. The two nodes are now able to network.

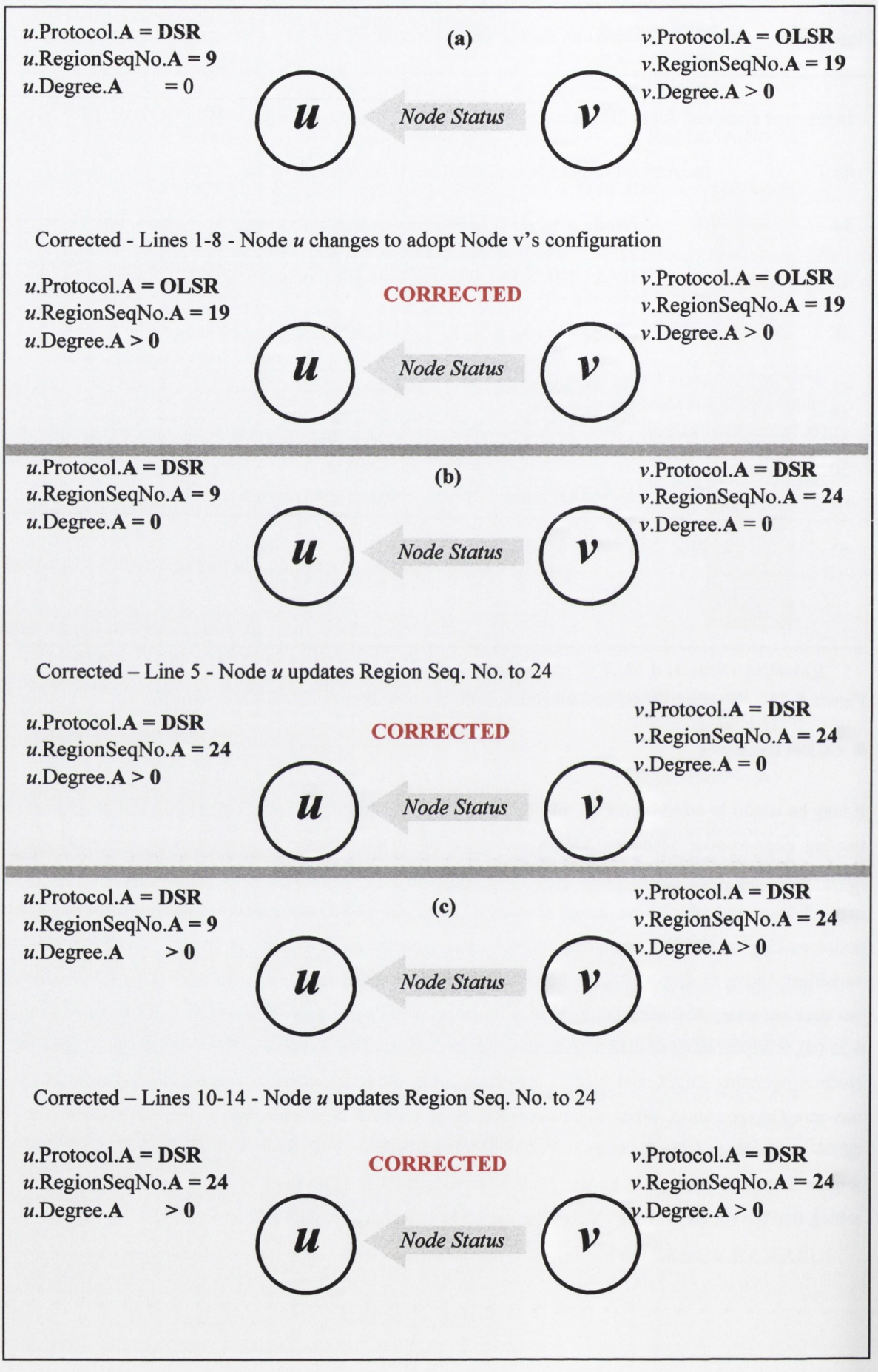

u.Protocol.A = DSR
u.RegionSeqNo.A = 9
u.Degree.A = 0
(a)
v.Protocol.A = OLSR
v.RegionSeqNo.A = 19
v.Degree.A > 0
u
Node Status
v
Corrected - Lines 1-8 - Node u changes to adopt Node v's configuration
u.Protocol.A = OLSR
u.RegionSeqNo.A = 19
u.Degree.A > 0
CORRECTED
v.Protocol.A = OLSR
v.RegionSeqNo.A = 19
v.Degree.A > 0
u
Node Status
v
u.Protocol.A = DSR
u.RegionSeqNo.A = 9
u.Degree.A = 0
(b)
v.Protocol.A = DSR
v.RegionSeqNo.A = 24
v.Degree.A = 0
u
Node Status
v
Corrected – Line 5 - Node u updates Region Seq. No. to 24
u.Protocol.A = DSR
u.RegionSeqNo.A = 24
u.Degree.A > 0
CORRECTED
v.Protocol.A = DSR
v.RegionSeqNo.A = 24
v.Degree.A = 0
u
Node Status
v
u.Protocol.A = DSR
u.RegionSeqNo.A = 9
u.Degree.A > 0
(c)
v.Protocol.A = DSR
v.RegionSeqNo.A = 24
v.Degree.A > 0
u
Node Status
v
Corrected – Lines 10-14 - Node u updates Region Seq. No. to 24
u.Protocol.A = DSR
u.RegionSeqNo.A = 24
u.Degree.A > 0
CORRECTED
v.Protocol.A = DSR
v.RegionSeqNo.A = 24
v.Degree.A > 0
u
Node Status
v

Figure 6-24 CCDR Examples - (a), (b) and (c)

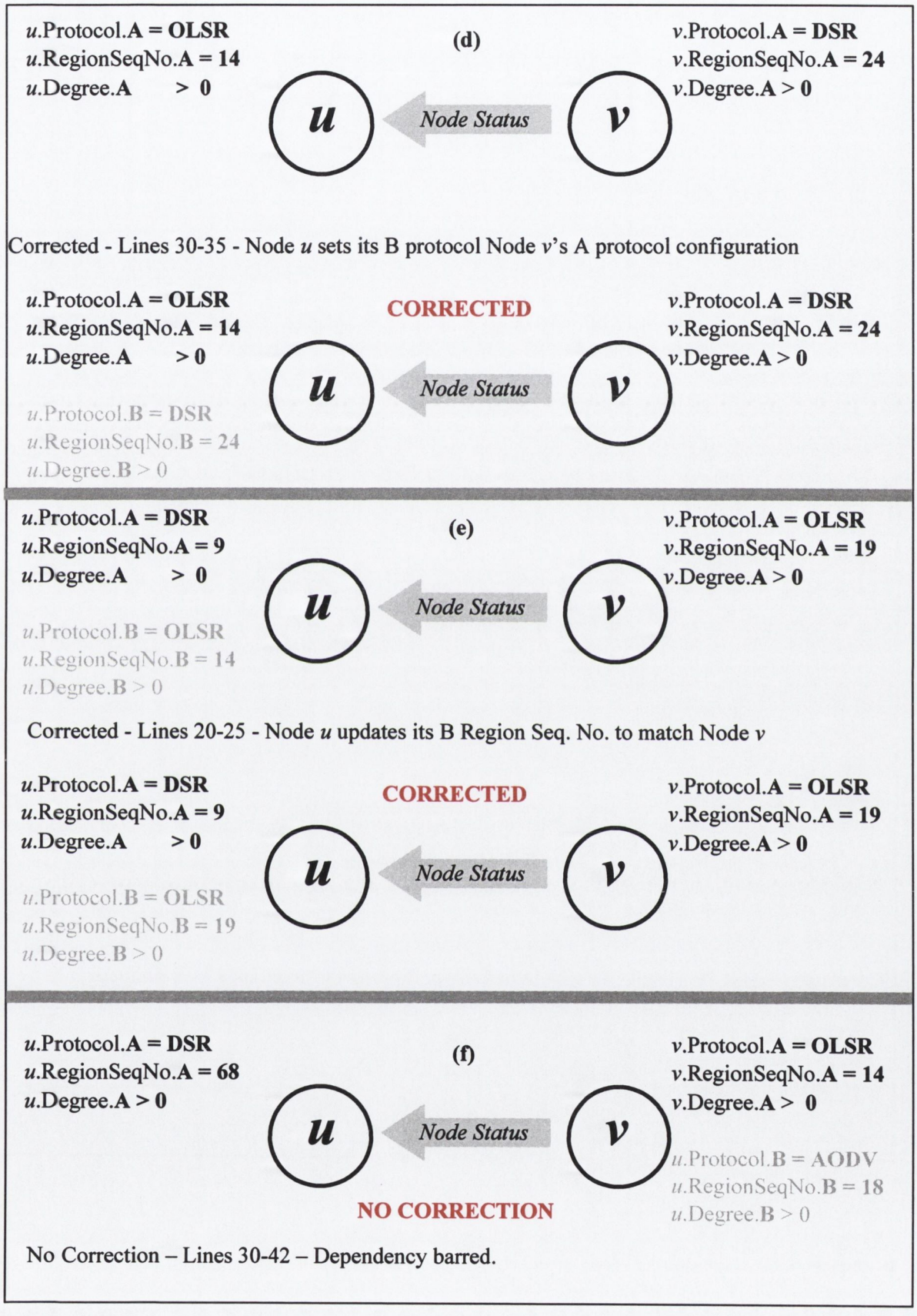


Figure 6-25 CCDR Examples - (d), (e) and (f)

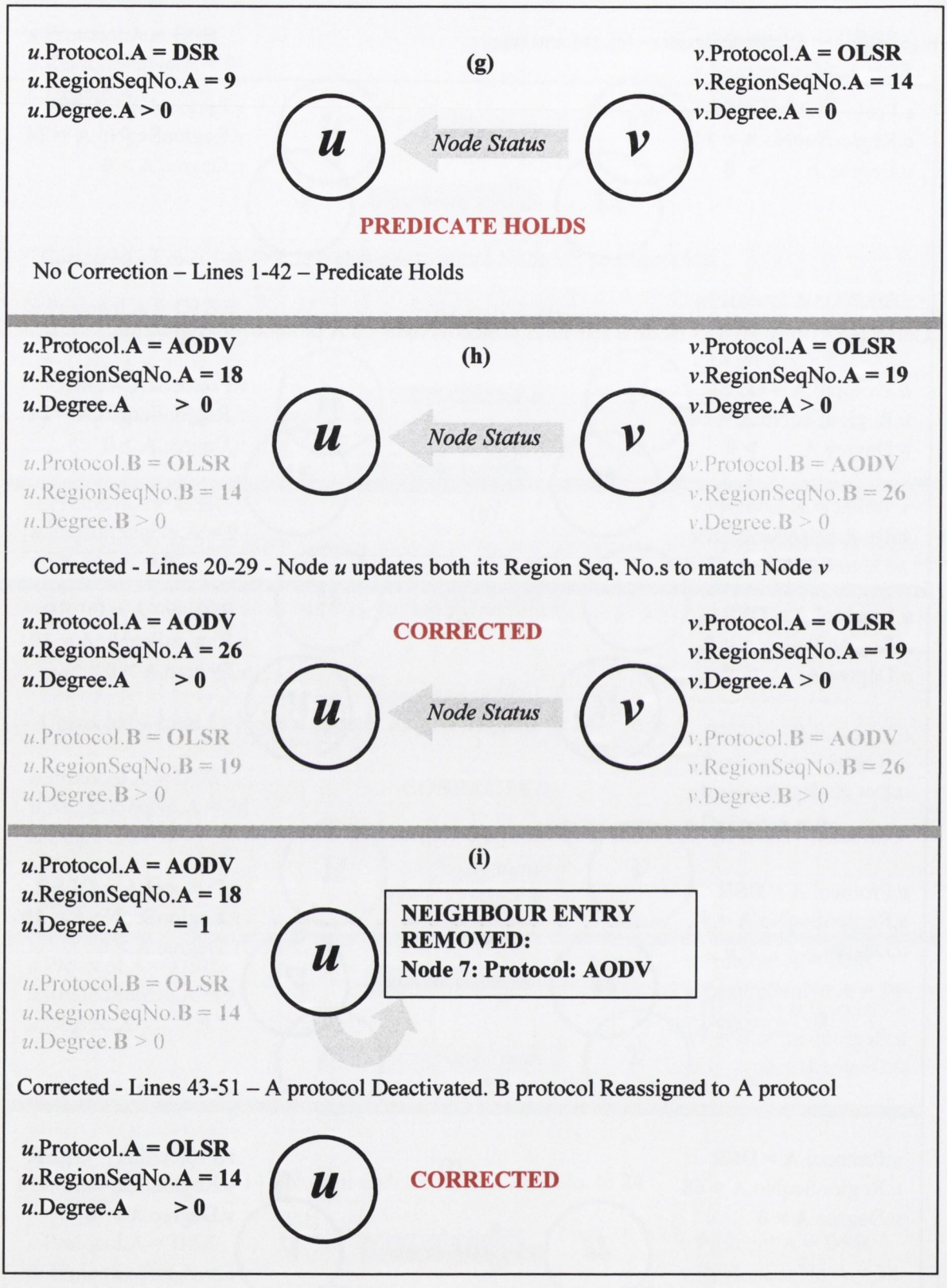


Figure 6-26 CCDR Examples - (g), (h) and (i)

■ **CCDR: Node Migration and Join**

The concept of node join and node migration was introduced and motivated in Chapter 3. Within the context of the operation of the auto-configuration protocol, node join and node migration occur during the course of

normal networking as nodes parse Node Status information according to the CCDR protocol rules. Node join and migration involves an isolated node, i.e. a node that has lost contact with its network, coming within transmission range of a network that has a different ad hoc routing protocol configuration. The node and the network may differ by both routing protocol and Region Sequence Number, or they may differ by Region Sequence Number alone.

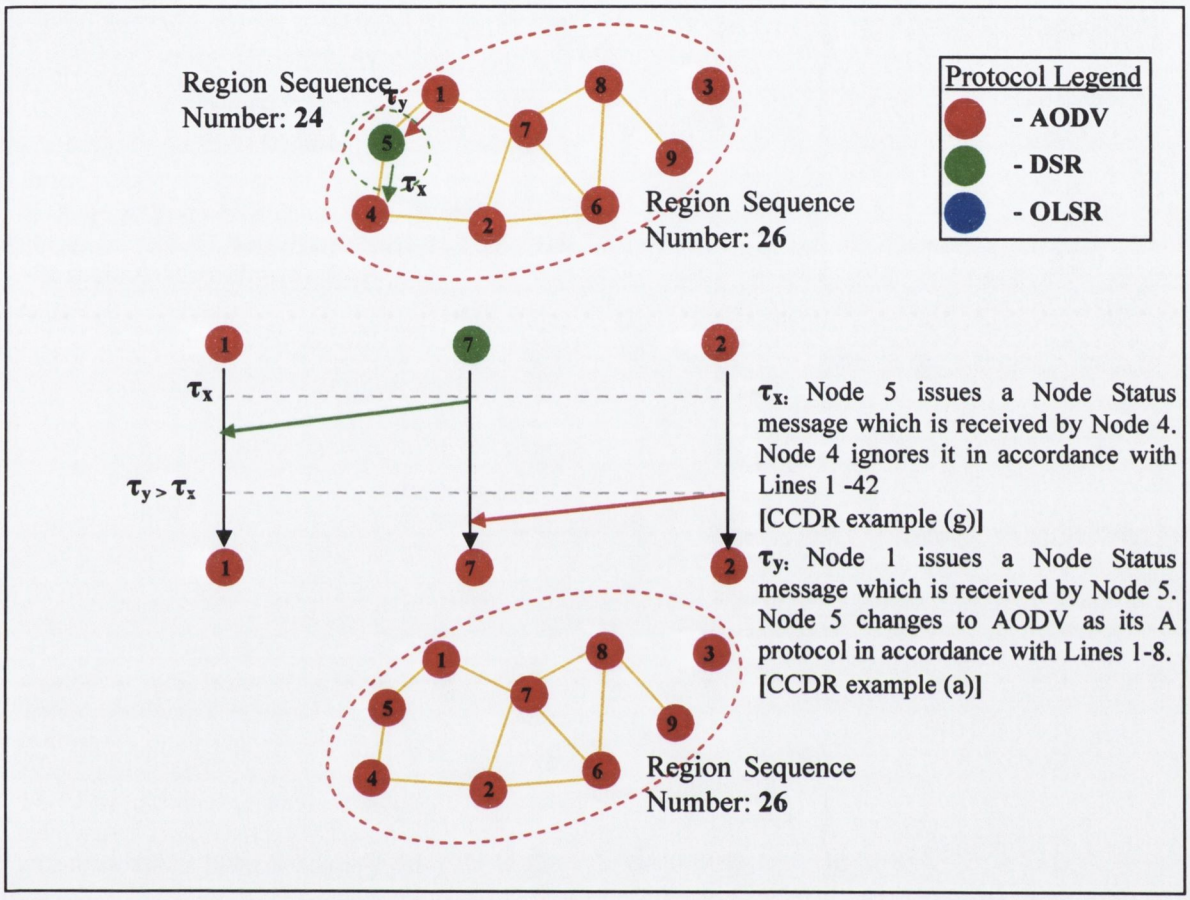


Figure 6-27 CCDR: Node Migration and Join

In the example depicted in Figure 6.27, an isolated DSR node has entered a region of connected AODV nodes. The signalling of the DSR Node is ignored in accordance with the CCDR protocol; a connected node does not adapt itself to suit an isolated node at any time. On the other hand, the AODV network is connected, which means that its nodes are signalling each other. When the DSR node receives a Node Status message at time τ_y, the DSR node unilaterally adopts the configuration of the AODV network in accordance with Lines 1- 8 of the CCDR protocol, thereby correcting the network state.

■ CCDR: Network Mergers

The more complex configuration conflict detection and resolution scenario arises during the merger of networks. Again, such disruptive networking scenarios were introduced and motivated in Chapter 3. A merger, using the meaning set out in Chapter 3, occurs when two heterogeneous networks, i.e. connected regions of nodes, come to occupy the same physical space and ether. The merger may come about if the edges of the two nodes meet or if the two networks either fully or partially overlap.

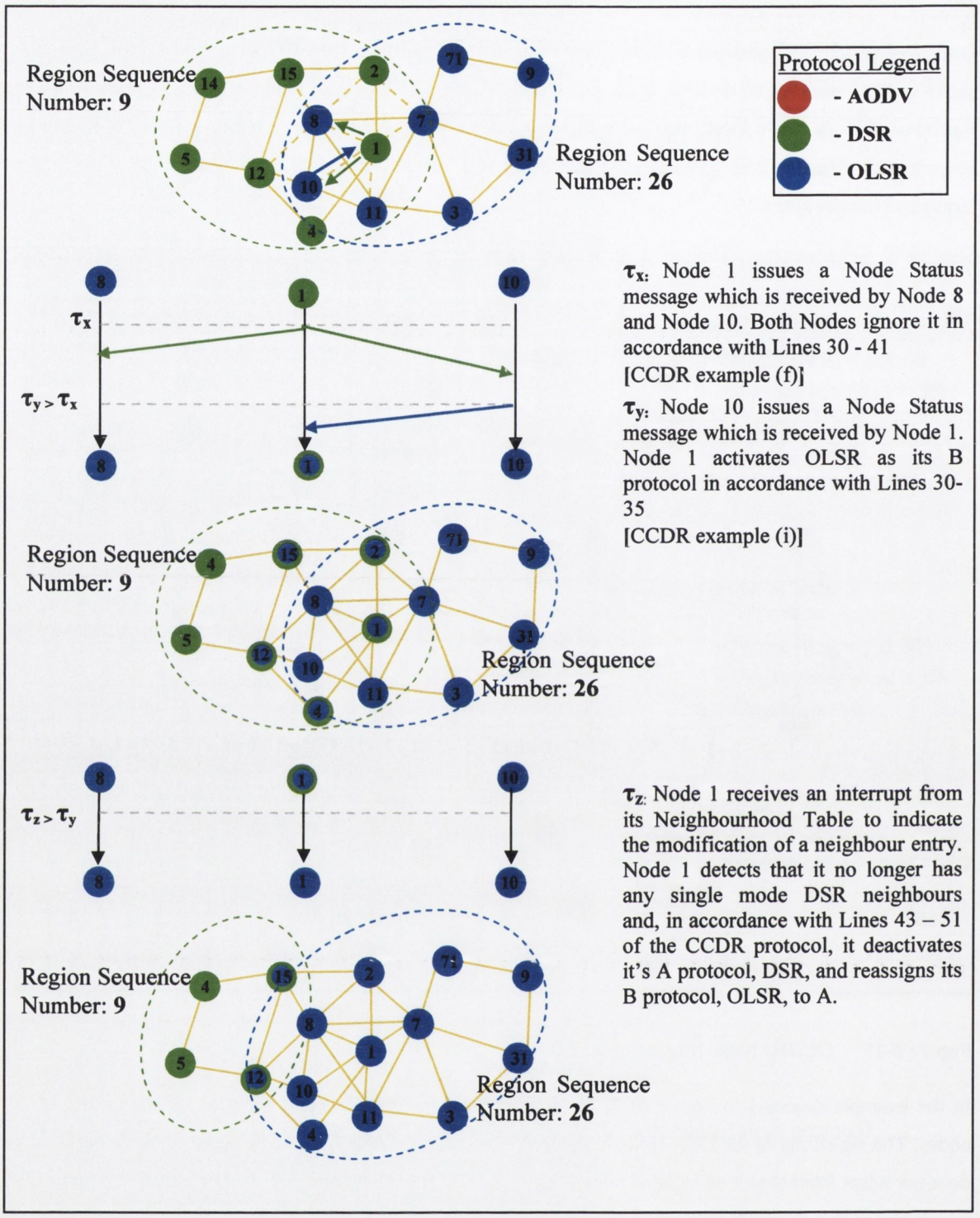


Figure 6-28 CCDR: Network Mergers

The example illustrated in Figure 6.28 depicts a partial merger scenario where an AODV region and an OLSR region have come to overlap the same physical space and ether. The overlapping region consists of a mix of nodes operating incompatible ad hoc routing protocols. While Node 1 is within transmission range of Node 7 they may not exchange multi-hop routing information; the same applies for other similar pairs of nodes in this example. As the nodes are actively signalling in order to maintain connectedness with neighbouring nodes, OLSR nodes will receive AODV signals and vice versa. At time τ_x, Node 1 issues a Node Status message which is received by OLSR nodes 8 and 10. Following the CCDR protocol, lines 30 –

42, these nodes ignore the Node Status message from Node 1. A time later at time τ_y, OLSR Node 10 issues a Node Status message which is received by Node 1. Node 1 yields to this message, in accordance with lines 30 – 35 of the CCDR protocol and activates it B protocol as OLSR.

The AODV nodes, Node 2, Node 3, Node 12 and Node 15, also go through the same process, although not necessarily at the same time as the protocol runs independently at each node and allows nodes within the same region to react asynchronously. Each node simply acts in accordance with the same CCDR protocol rules based on its own observations, i.e. received Node Status messages.

By time τ_z, Node 1 has received an interrupt from its Neighbourhood Table to indicate that a neighbour has been modified. Lines 43 to 51 of the CCDR protocol detect that Node 1's last single mode DSR-configured neighbour has been removed (it has either lost contact or the node has become a dual mode node) so it deactivates DSR, its A protocol and reassigns OLSR to its A protocol.

6.6 Summary

This chapter has focussed on enabling the auto-configuration of the network-layer's routing protocol. The challenges and objectives for such a component were identified and explored out with a view to developing a robust auto-configuration component. Following the design blueprint that was identified in Chapter 4, the various elements of the auto-configuration component were identified. These are based on the recurring need for observation, analysis and decision-making. The node's perspective on the network configuration was explored; a detailed characterisation of a node's ability to observe the network based on ambient signalling was presented. Such a characterisation clarifies the process of identifying and developing the various node properties that the auto-configuration protocol must anticipate.

As the component is concerned with the distributed organisation of a multi-state system of autonomous nodes, a detailed analysis of ad hoc management techniques, address auto-configuration schemes and the concept of self-stabilisation were presented in an effort to identify techniques for incorporation into the component. Having identified elements of the stateless address auto-configuration schemes and the technique of self-stabilisation by local checking and local correction as being suitable, a robust, extensible and low-cost auto-configuration protocol was then developed. The auto-configuration protocol consists of two sub-protocols, the Stateless Configuration (SCI) protocol and the Configuration Conflict Detection and Resolution (CCDR) protocol. The SCI protocol uses techniques analogous to the tentative stateless auto-configuration techniques used in the area of address auto-configuration. Coupled with the CCDR protocol, which incorporates robust and low-cost localised self-stabilisation techniques, the two sub-protocols ensure that the network, which is composed of collaborating multi-state autonomous nodes, is able to maintain a stable and legal global state with regard to the configuration of the nodes' network-layers. The next chapter presents a component which proactively goes about reconfiguring the network-layer's routing protocol in response to changed networking conditions in the context of the stable networking environment brought about by this auto-configuration component

7 Enabling Reconfiguration

7.1 Introduction

As outlined in Chapter 3, a key requirement of flexibility in a multimodal ad hoc network is that of being able to reconfigure the network-layer's choice of routing protocol. This chapter introduces and explores the issues involved in enabling groups of nodes to change their choice of routing protocol when the prevailing networking conditions undergo significant lasting change.

The following sections will clearly detail the objectives and challenges that motivate this feature of the flexible networking system. Then, returning to the design blueprint introduced in Chapter 4, the overall approach to the design of this component is laid out. As exemplified in the design of the auto-configuration component, one of the aims of the flexible multi-protocol network-layer is to be as sympathetic to the underlying fundamentals of the primary network-layer routing components as possible. To this end the design approach continues to eschew, in so far as it is possible, the use of techniques that induce extra signalling costs.

As the reconfiguration of the network reduces to the task of co-ordinating and organising the actions of a distributed group of nodes, a review and analysis of the area of distributed consensus is presented in Section 7.3. Distributed consensus is the fundamental concept underlying all distributed agreement problems. This review develops and furthers the discussion that was presented in the last chapter concerning ad hoc management schemes, address auto-configuration and self-stabilisation techniques. The review of distributed

consensus helps to highlight the limits of this concept and also helps to identify some of the fundamental elements of such a technique that must be incorporated in the design of the reconfiguration component.

It is apparent from the review of distributed consensus that the reconfiguration of nodes is not a trivial task; the mobile ad hoc network is a much more dynamic and heterogeneous environment than the typical distributed system cited in that literature. In order to further the design of the component it is necessary to introduce a more flexible decision model which takes account of the distinctive qualities of ad hoc nodes. Section 7.4 explores the concept of diffusion of innovations in an attempt to deduce a model or framework for the reconfiguration component which builds upon the observed characteristics of the participants in a societal innovation-decision-making processes.

Finally, Sections 7.5 and 7.6 present a protocol that addresses the problems which are associated with network-layer protocol reconfiguration by incorporating the more flexible approaches to distributed consensus with a modified version of the innovation-decision process. The Reconfiguration-Decision Protocol revolves around a continuous Persuasion Stage and a once-off Decision and Implementation Stage. The protocol enables nodes to make soft-state decisions according to a strict set of rules based on their network observations, their neighbours' observations and their neighbours' soft-state decisions. A hard decision is only implemented if the conditions persist.

7.2 Reconfiguration

7.2.1 Objectives

The objective of this component is to enable nodes to reconfigure themselves so that they are using the ad hoc routing protocol most suited to the prevailing networking conditions. The evaluation literature which was reviewed in Chapter 2 has comprehensively shown that various protocols are suited to disparate networking conditions. Whilst acknowledging the equivalence of the routing protocols, in that they all provide a multi-hop peer-to-peer routing service, it is apparent from the published literature that the efficiency with which different protocols deliver that service is dependent on the underlying networking conditions. It was also noted that the effects of local tuning of routing protocols' parameters has limited scope to ameliorate the ability of the protocols to handle diverse networking conditions. It is proposed in this thesis that a flexible multi-protocol network-layer should enable networks to utilise the optimum routing protocol for the prevailing networking conditions.

The literature shows that there is both a causal and correlational relationship between the local networking conditions, as particularly exemplified by node degree, node mobility and traffic loading, and the performance profiles of the affected nodes. The techniques described in Chapter 5 enable nodes to passively collate information regarding the significant local networking conditions. It was shown that these low cost metrics, which rely on the ambient network-layer signalling, capture and quantify the variation in the underlying networking conditions. Using this information, the aim of this component is to enable nodes to reconfigure the network-layer in response to significant long-term changes in the networking conditions, e.g. significant changes in the node mobility, node density and traffic loading so that the nodes are using the

protocol most suited to the prevailing conditions. As the network-layer is tasked with delivering as much data as possible (*maximal throughput*) using the least amount of network resources (*minimal control overhead*), the reconfiguration component is specifically tasked with optimising the choice of routing protocol with respect to these objectives.

7.2.2 Challenges

The design of this component is challenging on many fronts. Many of the problems associated with organisation of distributed systems and networks were described in the last chapter. Much like the auto-configuration component, the design of a reconfiguration component is challenged by every attribute of the proposed ad hoc network; specifically, it is challenged by the open, autonomous and distributed nature of the system, as well as the lossy and often asymmetric links.

Whereas the auto-configuration component enables nodes to take unilateral reactive corrective action when the state of the network is corrupted by naturally occurring networking events, the reconfiguration component seeks to change the routing protocol in use across a *body* of connected nodes in an organised and agreed manner. As such, the reconfiguration component must address issues that the auto-configuration component did not face. These issues centre around the network determining when and how to initiate, execute and terminate the reconfiguration process.

Furthermore, the open nature of the network means that nodes may enter or leave the network while the network is trying to reconfigure itself. The component must accommodate such population uncertainty and it must accommodate the unexpected departure of nodes participating in the reconfiguration process. The lack of synchronised global timing means that nodes cannot co-ordinate their actions based on a universal timeframe. Also, the distributed nature of the network inhibits communication between remote nodes, making co-ordination between distant nodes more difficult. It should be remembered that the system does not allow the flexibility-enabling components to make explicit use of the multi-hop routing services at the network-layer.

As the network may experience spatial and temporal variation in the networking conditions that it experiences, nodes may have competing objectives. The protocol must provide a means of reconciling the objectives of individual nodes and it must enable the network to decompose itself into multiple heterogeneous regions if necessary.

7.2.3 Design Blueprint

Returning to the blueprint of the flexible multi-protocol network-layer which was introduced in Chapter 4, the design challenges for this component can be summarised as finding mechanisms that enable nodes to reconfigure their choice of routing protocol so that the network they are forming uses the most suitable protocol based on observations and information made possible by means of techniques like those described in Chapter 5. The main elements of the component are highlighted in Figure 7.1. Once again, the successful operation of the component turns on its ability to observe the networking environment, analyse the observations and make co-ordinated decisions.

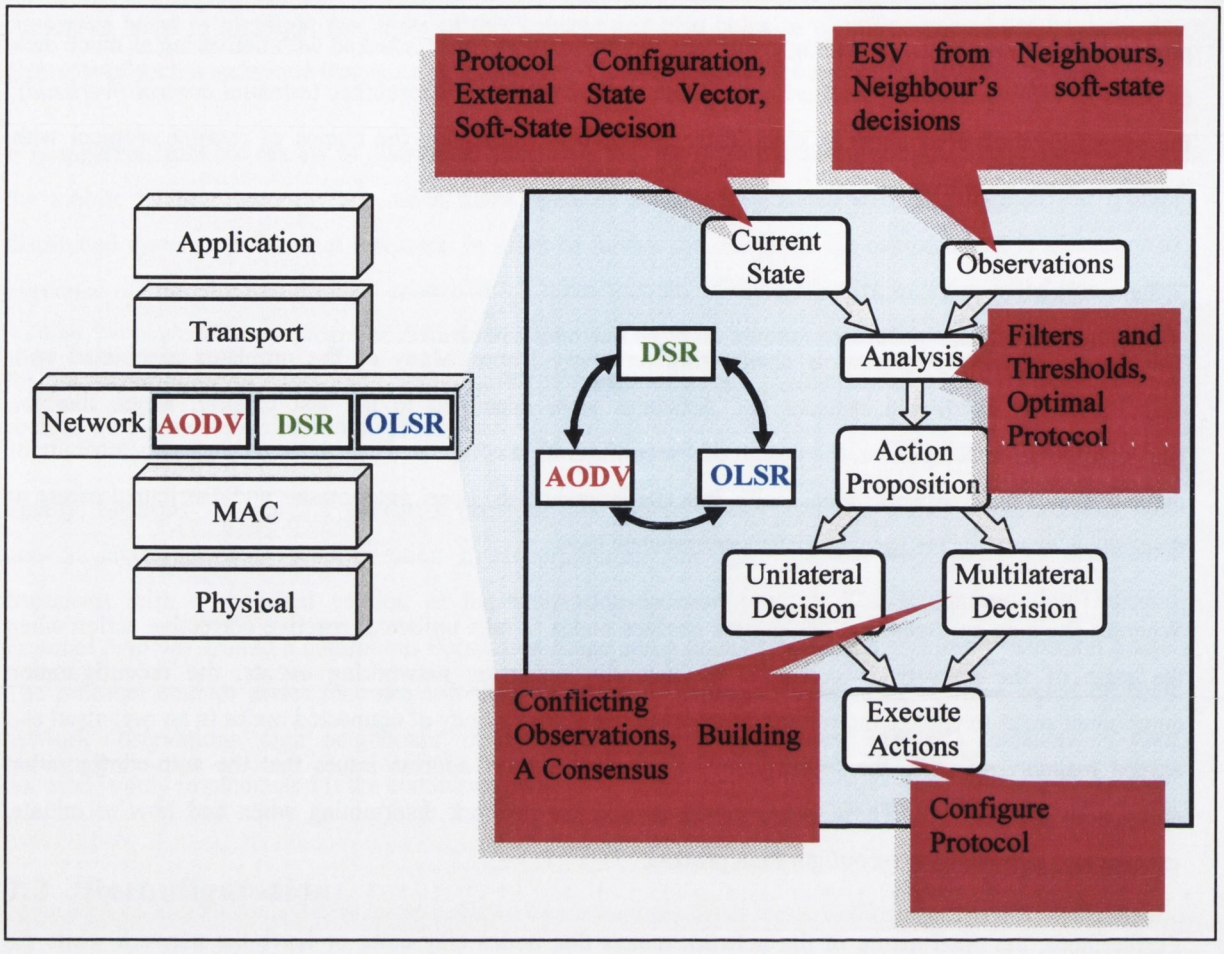


Figure 7-1 Blueprint for Design of Reconfiguration Component

The actions taken by a node will be informed by its current state, its network observations and the network observations of its neighbours. A node's basic state from the point-of-view of this component is its vector of collated myopic measurements which informs it as to the prevailing networking conditions.

The issues surrounding network observation have been discussed in both Chapters 5 and 6. In Chapter 5 it was shown that nodes can appreciate the prevailing networking conditions, while in Chapter 6 it was shown that a nodes' networking status does not have a detrimental effect on its ability to use ambient signalling to inform itself about the network state. Furthermore, it is shown that neighbouring nodes can share their observations in such a manner that allows nodes to collate a wider view of the local networking conditions.

As this component is tasked with allowing groups of nodes reconfigure their routing protocol, the techniques that will be used must allow for situations in which nodes have conflicting objectives for the reconfiguration of the network-layer. The component must enable nodes to build a consensus while also allowing for the innate characteristics of ad hoc nodes, i.e. their autonomous movement and lossy node-to-node links.

7.3 Distributed Consensus

In order to inform the design of a multi-lateral reconfiguration component it has been found useful to consider techniques and concepts that already exist in the literature. As noted in the last chapter, prior work

in the area of techniques that enable network-layer flexibility of the kind proposed and discussed in this chapter and in Chapter 3 is *very* limited. This review extends the discussion presented in Section 6.4 which focused on address auto-configuration techniques, ad hoc management schemes and the concept of self-stabilisation.

■ Distributed Consensus

Distributed consensus is concerned with the reaching of agreement among remote processes and is one of the fundamental co-ordination problems of distributed computing [Att98]. For the sake of consistency with the literature, the terms process or processor are used in this section in lieu of the networking terms node or host. In the consensus problem, a set of processes must co-ordinate their actions so that they *all* reach agreement on a decision based on their initial inputs which may be conflicting. Distributed consensus has applications in many areas. Among them are atomic broadcast protocols, transaction-commit protocols, the synchronisation of clocks, mutual exclusion and the election of leaders [Gue00]. In the case of the transaction-commit protocols which are widely used in distributed database systems, the decision may be binary in nature, i.e. either the remote processes 'abort' or 'commit'. In the case of mutual exclusion, processes agree on which one of them can enter the critical section. In any case, consensus protocols are not concerned with what decision is being made, only that all processes agree on the same conclusion. A broad review of consensus protocols for distributed systems may be found in [Tur92] which presents a discussion of possible applications of distributed consensus protocols over a range of different distributed systems which are characterised by varying fault, communication, transmission, message-ordering and processor assumptions.

To reach consensus, every process begins in an *undecided* state and *proposes* a value drawn from a set of possible values. The processes then communicate with one another to exchange their proposed values and then each process sets its *decision variable*, thereby entering the *decided* state. The correctness of a consensus protocol is characterised by the fulfilment of the following three conditions: *consistency of agreement*, *validity (or integrity)* and *termination*. In short, this means that; firstly, all processes agree on the same decision variable and that decision is final; secondly, that the agreed value must have been some process's proposed value and that lastly, each process sets its decision variable within a finite number of steps. Termination defines the *liveness* property associated with the consensus problem. i.e. this condition must eventually hold true. Agreement and validity are the *safeness* properties, i.e. these conditions must hold true at all times during the execution of the protocol. The agreement condition may be varied to allow for different types of decisions; in a uniform consensus protocol all processes must agree on the same decision variable but in a weaker majority agreement protocol only the majority of processes need to agree. It should be noted that consensus protocols require participating nodes to be aware of the number of other processes participating in the scheme in order to determine whether or not there is consensus or majority agreement. All other agreement problems may be shown to be at least as hard as consensus, not easier [Gue97]; therefore, a review of the distributed consensus problem highlights difficulties, constraints and opportunities that are present in all other distributed agreement problems.

Distributed consensus protocols may appear to be good candidates for use in distributed systems, such as mobile ad hoc networks, that require remote processes to agree on a common values or states that have global significance. Such a requirement may arise in a flexible ad hoc networking system when remote processes try

to reconfigure the network-layer by choosing a more suitable ad hoc routing protocol when the networking conditions have changed. While distributed consensus protocols have been successfully designed and deployed in systems that mirror the shared-memory model, distributed consensus protocols may not be as readily applied to the mobile ad hoc networking domain, as modelled by an asynchronous message-passing system. Distributed consensus algorithms must overcome fundamental difficulties that are introduced by factors such as asynchrony, limited local knowledge and failure. The effects of each of these factors are amplified in the mobile ad hoc networking context.

Most of the existing literature that describes solutions to the consensus problem assumes the existence of a network of reliable channels; a reliable channel is an abstraction that assumes that the underlying communication medium does not lose a single message. A reliable network provides services such as reliable ordered multicast and reliable broadcast which are fundamental to many distributed consensus protocols. The termination condition of a consensus protocol may be guaranteed by the reliability of the multicast operation as every process eventually receives the proposed decision value. In the context of a flexible unicast-based network-layer the assumption of reliable communication is unreasonable given that the network-layer does not normally offer reliable service primitives, merely a best-effort connection-less service. Reliable multicast communication is the preserve of higher service layers which may be built upon the basic service of multicasting protocols such as MAODV or ODMRP which themselves have different service objectives to unicast protocols such as AODV, DSDV, DSR and OLSR.

Many deterministic consensus protocols also assume the existence of a synchronous system, i.e. synchronous processes operating over synchronous communications systems. Synchronous systems assume that message exchanges take place in rounds and processes may make use of timeouts to detect other faulty processes. In a synchronous system a reliable channel eventually delivers each message, in spite of failures, within a specified time bound. However, it is acknowledged that real systems are asynchronous; they are subject to many possible faults such as network partitioning, asymmetric links, message duplication and message loss. In an asynchronous system processes can respond to messages at arbitrary times. It is also acknowledged in the literature that process clocks are unlikely to be synchronised. Within the context of a wireless mobile ad hoc network, the network-layer offers an unreliable time-free communication service in which messages may either be eventually delivered or lost.

The seminal contribution of Fischer, Lynch and Patterson proved the impossibility of distributed consensus under certain conditions [Fis85]. Fischer *et al.* demonstrated that deterministic distributed consensus is impossible in an asynchronous message-passing system if even one process crashes; a crashing process is the simplest form of system failure. This result is often called the FLP result. Impossibility results, such as the FLP result, are important as they inform us when a '*hard*' distributed problem has no solution. Much work has been carried out in an effort to identify other impossibility results that indicate where a designer should not put efforts into resolving a problem and a comprehensive study of such impossibility results is presented in [Fic03].

The basic reason why consensus cannot be achieved in a completely asynchronous system is that it is impossible to distinguish between a process that has crashed and one that is just very slow. Such a scenario is

typified by a mobile ad hoc networking system. The effect of asynchrony and failure occurrences actually creates an uncertainty, as perceived by a process, about the state of the system that can make it difficult, or even impossible, to determine a system view that can be validly shared by all non-faulty processes. The implications of this result are that there is no robust solution to this problem without making other assumptions about the computing environment. This has prompted the research community to explore mechanisms that can circumvent the FLP impossibility result for deterministic consensus in an asynchronous system. Some have suggested the use of randomised consensus or failure detectors or additional timing assumptions to get around the impossibility of deterministic consensus. Failure detectors are a practical solution to implement from the point-of-view of their simplicity and general applicability [Gue97]. Also, in the context of mobile ad hoc networking, the failure of processes is a common occurrence that must be overcome if processes are to run any type of consensus algorithm.

■ Failure Detectors

A failure detector is a distributed entity, or *oracle* as it is called in the literature, which provides the participating processes with a view of other processes that have crashed during the execution of the consensus protocol [Cha96]. Failure, as defined by processes crashing, would be a common feature of a mobile ad hoc network where nodes may leave or fail unannounced. Byzantine (or arbitrary) failure is not considered in this thesis, nor is it considered in the design of failure detectors. Byzantine failure may be countered by demanding that messages between processes are digitally signed. A failure detector helps a consensus protocol to operate in the presence of faulty or slow processes by helping the system to distinguish between such processes. Failure detectors are not necessarily accurate, especially if they operate in an asynchronous environment and they may be classed as either *reliable* or *unreliable*.

Chandra *et al.* present the concept of an unreliable failure detector for an asynchronous system susceptible to crash failures [Cha96]. Unreliable failure detectors classify processes as either *suspected* or *unsuspected,* the detectors cannot be certain as to whether a process has actually failed or not. Failure detectors will be unreliable as they themselves may make mistakes if they are based on message-passing in an asynchronous system, a property that is consistent with the earlier FLP impossibility result. The failure detector is manifested in the system at each process by a local failure detector module that monitors a subset of the processes in the system. If a process is suspected it means that the failure detector has, perhaps, not received a message from the process within some time limit. Even though the system is asynchronous, practical time limits may still be used to infer hints about the status of the process. If a process is unsuspected it means that the detector module has recently received some information, perhaps a message, to suggest that the process has not failed. The response that a failure detector gives to a process about the state of other suspected or unsuspected processes is only as good as the information at that process which is dependent on the communication conditions. As these conditions may vary from process to process, the failure detector may give different responses at different processes.

An example of an unreliable failure detector, q, may be implemented using a heartbeat mechanism. Each process, p, sends a 'p is alive' message to every other process every T seconds, akin to a hello message in a network protocol. The failure detector estimates the maximum message transmission time of D seconds. If a 'p is alive' message is not received in $D + T$ seconds then q reports that p is suspected. If p then sends a 'p is

alive' message q reports that p is unsuspected, i.e. process p is OK. In practical systems there are bounds on maximum transmission times, whether the system is asynchronous or not. If the value chosen for $D + T$ is too small then a failure detector is likely to suspect a lot of non-crashed processes regularly as the window to receive 'p is alive' messages is too small for the underlying system properties. A small value for $D + T$ may also result in the 'p is alive' messages wastefully using up bandwidth. On the other hand, if the value chosen for $D + T$ is too large then crashed processes may go unsuspected by the detector. In a practical system, a solution is to tailor the timeout values to the observed network delays.

It may be observed that the heartbeat mechanism is similar to neighbour-sensing processes used in a proactive ad hoc routing protocol such as OLSR. OLSR uses periodic link sensing messages, i.e. 'p is alive' messages, and topology update messages to enable distributed nodes to build a view of nodes that are alive in the system. OLSR sends messages every T seconds, where T is based on the link transmission time estimate for the underlying system. However, such control signalling is not present in reactive protocols which are purposefully quiet when there is no demand to route traffic. Significantly, failure detectors which observe the state of processes that are not within direct communication range of the detector itself depend on the existence of multi-hop communication facilities.

When a failure detector is being used, the processes involved need to be modelled as either *crash/recovery* processes or *crash/no recovery* processes. Failure detectors generally assume the former model. The main difference between two such processes is that the crash/recovery process has access to persistent memory that allows it to pick up where it left off when it crashed. In some cases a crash/recovery process model is needed as consensus may not be realised in the system if the recovered process cannot partake in the consensus execution. For example, consider a system of three processes executing the j^{th} run of a majority agreement protocol. If one of the processes crashes and recovers during that run of the protocol, but is not allowed to participate in the run being executed, then the consensus process may deadlock if the remaining processes do not agree, i.e the two remaining processes cannot form a majority agreement. In the case of the crash/no recovery process model a process may learn of the result after agreement has been reached by the remaining uncrashed processes. Such a process is allowed to participate in future runs of the protocol, i.e. the $(j+1)^{th}$ run of the protocol.

As a network may experience failure when nodes crash or when nodes temporarily move out of communication range it is necessary to consider both types of models. For the case of a node failing by crashing, such a node may follow the crash/no recovery model as a node will generally not have any persistent memory. In the case of a node failing through transient link failures the node may follow the crash/recovery model depending on the type of agreement scheme implemented. In practical systems, processes can deem another process that has not responded for more than a bounded time to have failed. While the process may not have really failed, the other processes can act as if it did. Such failure detectors allow a distributed consensus algorithm to make a process failure '*fail-silent*' by discarding any future messages that they receive from the 'failed' process for the rest of the consensus execution time. This has the effect of turning an asynchronous system into a synchronous one.

■ Group Membership

One area in which the concept of distributed consensus has been applied to ad hoc networks is in the development of *group membership* schemes. Group membership schemes are of interest to the ad hoc networking community as they enable spontaneous communities to cooperate at application level, mirroring the ability of nodes to internetwork at the network-layer. Group membership is essentially an agreement problem as processes must agree on the system view. A review of group membership schemes for asynchronous systems is presented in [Gal96]. Group membership services provide a middleware solution to the designers of distributed applications. A group membership service maintains a consistent *view* among correct processes of the set of correctly functioning processes that cooperate to provide a service. The group membership scheme also provides information on specific state information, such as the queues of accepted requests, requests not yet completed and the state of the resources used to provide the service. Many distributed applications rely on the notion of membership, i.e. each node maintains an estimate of which other nodes participate in the computation and are alive and connected. Generally, group membership protocols require nodes to have heartbeats indicating whether a node is dead or alive; again this is akin to the hello messages used in proactive routing protocols for link sensing or the messages used by failure detectors.

While many of the group membership schemes have been designed to deal with varying degrees of node mobility, Briesemeister *at al.* [Bri02] suggest a scheme that may be suitable for mobile ad hoc networks which is based on the notion of local group membership; other mobility-tolerable schemes rely on location information [Rom01], [Pic99]. The authors acknowledge the work of Chandra *et al.* [Cha96b] which shows that group membership is impossible in a single partition network. Single partition membership services allow only one network component to run the service as processes in other network partitions are considered faulty. Chandra *et al.* [Cha96b] note that group membership is a weaker form of the consensus problem. The approach suggested in [Bri02] considers partitionable group services which allow for multiple disjoint views to exist concurrently in different network components, i.e. processes and partitions. The authors argue that a service designed for mobile ad hoc networks should be partitionable as disconnections may occur frequently. Their localised group membership scheme allows an unbounded number of processes to exist. The nodes participating in the group need not be aware of an upper limit on the population size. It is also accepted that a mobile ad hoc network will characteristically experience lossy transmission, a point rarely acknowledged in other literature. The localised group membership service is built around a neighbourhood service that employs a heartbeat mechanism to enable processes to produce a list of neighbouring nodes that are expected to be alive. The heartbeat operates in an asymmetric manner so that neighbouring processes do not have to have to share a symmetric link, an important attribute in the context of wireless networking. This application of the distributed consensus technique serves to illustrate the narrow scope within which consensus has been applied to mobile ad hoc networking systems. In essence, the scheme devised in [Bri02] is little more than a very reliable neighbourhood detection service.

■ Summary

Notwithstanding the possibility or impossibility of applying a distributed consensus protocol to an asynchronous system, such protocols are also characterised by their message complexity which may be polynomial or exponential with regard to the number of processes involved. This characteristic is important

when considering the applicability and scalability of such a protocol to the wireless mobile ad hoc network. The literature on distributed consensus algorithms largely focuses on system models that bear little resemblance to that of the wireless mobile ad hoc network and consequently they are often modelled by totally connected networks in which reliable multi-hop communication is not an issue. While certain characteristics of the wireless mobile ad hoc network are represented in the literature, the incorporation of these characteristics into the system model is generally the exception rather than the rule.

Key Stages in Consensus

In spite of any of the technical hurdles to realising a distributed agreement solution in an mobile ad hoc networking system, it is apparent from the literature that there a number of key stages in the construction of an agreement protocol that must be incorporated into any practical solution. Firstly, a proposal must be initiated by some process suggesting that an action be taken or that a state be assumed by the participating process; in short, a decision variable must be proposed. Then, a period of deliberation or consultation takes place, which may be iterative, in which processes make a decision. Finally, the decision-making process must terminate when the participating processes either accept or reject the proposal.

Soft-state Termination

The issue of terminating the decision-making process within the context of a mobile wireless ad hoc network has been addressed in part by the proposal made in the MANETconf [Nes02] address auto-configuration scheme; MANETconf exemplifies the difficulties faced by distributed agreement schemes in a mobile ad hoc network. This scheme adapted the Ricart-Agrawala mutual exclusion algorithm [Ric81] so that it could overcome some of the hurdles inherent in an ad hoc network. In short, the MANETconf scheme employs a soft-state approach which allows a node to take an address while the mutex-based allocation process is ongoing. If the attempted allocation is denied to the node, or if the node does not receive a confirmation message within a fixed period, then the node relinquishes the address. This approach guarantees termination of the process in the presence of lost or delayed confirmation messages. The use of soft-state decisions and passive confirmation techniques in an ad hoc network may overcome difficulties in assuring the termination of an agreement scheme. However, unlike the MANETconf scheme, a decision-making process for reconfiguration cannot rely on explicit multi-hop routing services of the nodes participating in the network to deliver messages to other non-neighbouring nodes.

Local Decisions

Another hurdle, which was briefly referred to earlier, is that consensus is required among all of the processes in the system, implying that the number of processes is known. Even in the less stringent case of majority agreement protocols, the number of participants must be known so that the definition of what constitutes a majority within the system may be ascertained. In terms of the aims of the reconfiguration component, a number of observations can be made about the possible application of distributed consensus to the entire network. The main observation is that the reconfiguration of an 'entire' network is not possible as it is not possible to appreciate what constitutes the entire network. Nodes can only be assumed to have local knowledge pertaining to their neighbourhood which is dependent on the activity of the signalling profiles of

the neighbouring nodes. For information regarding the state of the network beyond the nodes' immediate neighbours, such information may be found in the route cache of the nodes. However, it has been noted that the extent or completeness of such information is dependent on the underlying routing protocol. In any case, regardless of the protocol in use, the information concerning the total node population, as provided by the route cache, is transient, lossy and unreliable.

In sum, it has been shown in the literature that is only possible to maintain a correct view of local processes. The local group membership scheme [Bri02] cited above provides an illustration of the fact that local decisions are the only ones possible in an independent ad hoc network.

Finally, while it is possible to circumvent the failure of processes during the execution of a round of decision-making, such a process may subsequently recover, or connect to the system, after a decision has been taken and acted upon. In the context of a reconfiguration decision, this may result in a node being left out-of-step with its neighbouring nodes once it regains a connection to the network.

7.4 Diffusion of Innovations: From Innovators to Laggards, Knowledge to Implementation

In designing a multilateral reconfiguration agreement protocol within the bounds defined by distributed consensus literature which were outlined in the previous section, it is necessary to tease out the characteristics of the participating nodes. In particular, since the mobile ad hoc network consists of autonomous nodes with varying networking experiences, it is useful to categorise these different types of ad hoc nodes with respect to the part that they would play in the reconfiguration decision-making process. To this end, this section presents a discussion of pertinent elements of the concept of *diffusion of innovations*; a detailed review of this social-science concept is presented in Roger's oft-cited publication [Rog96]. While there are other socio-economic models which attempt to describe this phenomenon, it will be shown in the next section that some aspects of the diffusion of innovations model promoted by Rogers may be used in the design of a reconfiguration protocol. The veracity of the model is not at issue in this work.

■ Introduction

The concept of diffusion of innovations arises in the study of social and cultural behaviour with regard to how innovations come to be adopted or rejected by members of a society. Simple examples of various forms of this concept range from the adoption of the QWERTY keyboard standard over the ergonomically superior DVORAK keyboard to the adoption of citrus fruits by the British Navy in the fight against scurvy from the seventeenth to the nineteenth century. Nowadays, non-governmental organisations, which are described as change agents in the literature, are often concerned with the diffusion of new agricultural techniques or disease-control techniques in developing countries. An innovation is a new idea that *may* bring some sort of benefit to the adopter.

At this stage a simple parallel is drawn between the concept of diffusion of innovations and the reconfiguration of the multi-protocol network-layer. Within a social context, human adopters undergo an innovation-decision process in which they learn of an innovation and come to a decision as to whether or not

they will adopt it. The innovation-decision process is characterised by the channels over which individuals learn of the innovations, whether they perceive a need for them and what their disposition is towards change. Innovation, in the context of this thesis, is an analogy for change at the network-layer, specifically a change in the ad hoc routing protocol when the need arises due to significant and lasting changes in the networking conditions. Ad hoc nodes exhibit many of the same characteristics of individuals involved in an innovation-decision process. Nodes have varying degrees of connectedness to other nodes and varying degrees of experience of the network. The innovation may be seen as the idea to adopt a routing protocol that will improve the efficiency of the participating nodes.

Observation of the diffusion of innovations in real life has led researchers to fit a model to those observations. The model consists of characterisations of the participants in the diffusion process as well as definitions of the various stages and elements of the innovation-decision process itself. The literature defines diffusion as the process by which innovation is communicated through certain channels *over time* among members of a social system. It is a special kind of communication in which the exchanged messages are concerned with new ideas. Communication is the process in which participants create and share information with one another in order to reach a mutual understanding. This definition implies that communication is a process of convergence (or divergence) as two or more individuals exchange information in order to move towards each other (or away from each other) in the meanings that they ascribe to certain events. This is what happens in a process of diffusion such as when a change agent seeks to persuade a client to adopt an innovation.

The literature distinguishes between centralised and decentralised diffusion. In the case of centralised diffusion, decisions regarding such matters as when an innovation should be diffused, how it will be diffused and how it should be evaluated are decided by a small number of individuals heading a change agency. In the decentralised diffusion model, which is patently of more relevance to this work, such decisions are shared by the clients, the potential adopters. In extremely decentralised systems, there may be no change agency and the potential adopters are solely responsible for the self-management of the diffusion of innovations. This form of diffusion relies on communication through what are referred to in the literature as horizontal networks, a concept somewhat analogous to a distributed ad hoc network. New ideas, or innovations, may grow out of the experience of certain individuals, or potential adopters, rather than through the specific promotion of centralised change agents.

■ Innovation-Decision Process

The innovation-decision process is the process through which an individual or decision-making unit passes from first knowledge, or awareness, of an innovation to forming an attitude towards the innovation, to deciding to adopt or reject it, to implementation of the new idea, and to confirmation of this decision. As the process happens over time, the five steps in the process are individually conceptualised as; knowledge, persuasion, decision, implementation and confirmation. Exceptions to this sequence of events may occur; this model is built around observations of many Western cultures and variations have been noted in its application to societies with different cultural values.

An innovation can either be adopted or rejected by individual members of the system or the decision can be made by the social system which can decide to adopt an innovation by collective or authoritarian means. These notions of decision-making gives rise to the terms *optional*, *collective* and *authority* decisions. Optional innovation-decisions are made by an individual independently of the decisions made by other members of the system. Nonetheless, the decision is still influenced by the system and interpersonal connections. Collective innovation-decisions are made by consensus among the members of a system. Each unit in the system must conform to the system's decision once it has been made. Authority decisions are made by relatively few individuals in the system who have power, status or expertise. The individual member has very little or no influence on the decision and must follow it once made.

Before describing the five stages of the innovation-decision process, a short description of the diffusion channels and sources is presented. The communication channels are, as with most other concepts, the means by which the messages get from the source to the receiver. Channels are either interpersonal or mass media in nature. Messages originate from either localite sources, i.e. those which are parochial in outlook, or cosmopolite sources. The two channels play different roles in creating knowledge of an innovation versus persuading an individual to form an attitude towards an innovation. As may be anticipated, a mass media channel allows the few to reach the many in order to spread information, but may only lead to awareness among the many and possibly changes in the few with weakly held attitudes. Interpersonal channels involve direct face-to-face (or neighbour-to-neighbour) exchange between two individuals. They allow for persuasion and are best when two-way exchanges are possible. Cosmopolitan sources are found to be more important at the knowledge stage and localite channels are more important at the persuasion stage. Cosmopolite sources are normally outside the interpersonal social system whereas localites are those within the social system.

Knowledge, the first stage in the innovation-decision process, occurs when a decision-making unit is exposed to the innovation's existence. Within the diffusion of innovation research community there is debate as to whether knowledge arises out of exposure to, or awareness of, the innovation or whether knowledge is driven by need. Need is driven by an individual being in a state of dissatisfaction that arises when one's desires outweigh the actualities, i.e. when the individual's current technology or technique or product does not provide the service that the individual desires. The innovation may be seen as a way to meet this desire, so when an individual forms a need it then seeks awareness of an innovation. In more artificially contrived situations, need may arise when a change agent makes an individual aware of an innovation through mass media messages, leading to possible individual dissatisfaction.

Persuasion occurs when the decision-making unit forms a favourable or unfavourable attitude towards the innovation. Mass media messages are generally not enough to provide reinforcement in the persuasion stage; individuals seek reinforcement from personal connections and consider the innovation in light of their current situation. As the innovation carries a degree of risk, individuals want to know that their thinking is in line with that of their peers so that it reduces uncertainty about the innovation. At this stage, an individual may be selective with regard to where it seeks information or how it receives and interprets that information. During this stage the adoption or rejection of an innovation is not a foregone conclusion, but there is a tendency for attitudes to become more consistent during this stage.

Decision occurs when the decision-making unit engages in activities that lead to either the acceptance or rejection of the innovation. Adoption is a decision to use the innovation, rejection is a decision not to adopt the innovation. Decision-making units or individuals will often trial an innovation on a probationary period at this stage as part of the decision to adopt. However, in many cases it is not possible to trial the innovation alongside the status quo and so the innovation must either be accepted or rejected in its totality. However, in some cases and for some innovations, the trial of a new idea by a peer can be a substitute for their own trial; such 'trial by others' provides a vicarious trial for the individual during this stage. In addition to their trial-ability, innovations are also characterised by the relative advantage they offer, their compatibility with the needs of the potential adopter, and the operability of the innovation, i.e. the degree to which the results of the innovation are visible to others.

An innovation can be rejected at any stage of the process, e.g. it can simply be ignored at the knowledge stage and forgotten about. At the decision stage, the literature distinguishes between two types of rejection. Passive rejection occurs when a node never really considers adopting the innovation. Active rejection, on the other hand, occurs when a node does consider the adoption of the innovation but then rejects it. This consideration may include the trial of the innovation, whether trialled by the individual or by others.

Implementation occurs when a decision-making unit puts the innovation into use. This stage normally follows directly after the decision stage within a very short period of time. The innovation finally loses its distinctive quality as the separate identity of the new idea disappears. This stage often represents the end of the decision-innovation process for many individuals.

Confirmation then occurs when a decision-making unit seeks reinforcement of an innovation that has already been made, but the unit may reverse the decision if it is exposed to conflicting messages about the innovation. At this stage, the individual seeks to avoid a state of dissonance or to reduce it if it occurs. If the individual is in a state of internal disequilibrium or dissonance, then the individual would ordinarily seek to eliminate or reduce it through discontinuance of the innovation. Discontinuance is a decision to reject the innovation having previously adopted it. It takes the form of a replacement discontinuance, which occurs when an individual rejects an idea in order to accept a better idea that supersedes it, or a disenchantment discontinuance, which occurs when an individual rejects an idea because of dissatisfaction with it and moves back to the previous state. In designing a reconfiguration component, the notion of replacement discontinuance fits closely with the concept of continually adopting a new protocol when circumstances dictate such action, i.e. the stated aim of such a flexibility-enabling component. The notion of replacement discontinuance suggests that the innovation-decision process is a continual and cyclical process in which the system will look for another innovation if it fails to receive reinforcement about the most recently adopted innovation. The flexible ad hoc networking system has the same objective of constantly observing and adapting and then observing again, and so on.

■ From Innovators to Laggards

The literature goes on to provide neat characterisations of the actors in the innovation-decision process using *ideal types*. Within the diffusion of innovation research field these ideal types are used to guide research

efforts and to serve as a common comparison framework for the exposition of research ideas and findings. Ideally, the categories should be exhaustive, i.e. they should include all types of decision-making units in the study, they should be mutually exclusive and they should be derived from the one classificatory principle. The most popular classificatory principle is based on the innovativeness of the decision-making unit. The literature describes innovativeness as the degree to which an individual or unit of adoption, e.g. a node, is relatively earlier in adopting new ideas than other members of the system. Five adopter categories are defined; innovators, early adopters, early majority, late majority and laggards.

Innovators are generally active information seekers about new ideas; in terms of sources, they are cosmopolites. They have a high degree of exposure to pertinent information and are eager to try new ideas. The innovator is willing to take a risk on change, accepting that not all attempts at innovation will be successful. However, while the innovator's venturesomeness is important in kick-starting the diffusion process, it does not play an important role in the persuasion of others to adopt the innovation due to its risk-taking profile. Nonetheless, the innovator does have the important role of acting as the gatekeeper that controls the introduction of innovations to other individuals.

Early adopters have the largest degree of persuasive opinion leadership in most social systems; in terms of sources, they are localites. Early adopters have a high degree of connectedness and make reasoned decisions. They serve as models for later adopters. The early adopter has to be more judicious in its innovation decisions in order to continue to command the respect of others. They decrease local uncertainty about an innovation by adopting it and conveying messages to other near-peers by means of interpersonal networks.

The *early majority* forms a bridge between the early adopters and the late majority. They may deliberate over the adoption of the innovation for a relatively longer period than either the innovator or the early adopter. The early majority do not lead opinion as the early adopter does, but through frequent interaction with their peers, they tend to follow the early adopters. They are crucial in terms of tipping the balance within a society towards adoption of the innovation. They provide an interconnectedness between groups of early adopters.

The *late majority* are often forced to adopt the innovation as a result of economic necessity and increasing social pressures. The late majority, being sceptical, wait until most of their social system have already adopted, i.e. the late majority are more influenced by peer pressure than the utility of the innovation. The late majority adopt an innovation after the average member of the social system has already done so.

Laggards, as the name suggests, are the last in a social system to adopt an innovation. They have no opinion leadership qualities and are very local in outlook; in fact many are near isolates in social networks. The point of reference for laggards is the past, their decisions are made in terms of what has been done in previous generations. Laggards are often lagging far behind in their awareness of a new idea. Laggards are prone to disenchantment discontinuance as they tend to fall into using an innovation after the more reasoned stages of the innovation-decision process have ended.

■ Critical Mass and the Heterophily of Individuals

There are two further points that are worth noting about this theory. The first concerns the concept of critical mass; the second revolves around the heterophily of interacting individuals. Critical mass occurs at the point at which enough individuals have adopted an innovation so that the rate of further adoptions becomes self-sustaining. Critical mass is important in the diffusion of interactive innovations where each additional adopter increases the utility of the adoption for all adopters. Reaching the point of critical mass involves persuading the early adopters to adopt the innovation after the innovators have introduced the new idea to the system. An interactive innovation is one for which the utility of the innovation rises as the number of near-peer adopters also rises. For an interactive innovation, such as a new routing protocol, until there is a critical mass of adopters, there is little advantage in adopting the innovation. After the critical mass has been reached the social system encourages further adoption by individual members of the system. In the case of an interactive innovation, individuals are likely to watch the other members of their group in an effort to discern what the group choice may be. The outcome for the group then turns on each member watching what the other members will do.

Accompanying the notion of critical mass at the societal level is the concept of a threshold at the individual's level. A threshold is the number of other individuals who must have adopted an innovation before a given individual will do likewise. In the case of the diffusion of an innovation, a threshold is reached when an individual is convinced to adopt as a result of knowing that some minimum number of other individuals in the system have also adopted. So, the reaching of critical mass turns on the actions of individuals with respect to their own thresholds.

The final point of interest taken from this theory concerns the homophilousness of the interacting individuals. The interaction between individuals and its impact on the diffusion of innovations between them is affected by the similarity of the individuals. Homophily, specifically in this context, is defined as the degree to which pairs of individuals who interact are similar in attributes. In societal terms, homophily is exemplified by attributes defined by beliefs, education and social status. In free-choice situations, an individual has a stronger tendency to interact with someone who is like them. The effects of homophily can be explained by the fact that similar individuals normally belong to the same groups, live or work near each other, and have the same interests. The physical and social closeness of such individuals makes homophilous communication more likely and such communication is normally quite effective. When individuals share common meanings and a mutual subcultural language, their communication of ideas is likely to have a greater effect in terms of sharing knowledge, forming attitudes and fomenting change. However, the diffusion of innovations, by its very definition, can only occur between participants who are somewhat heterophilous with regard to their grasp of an innovation, i.e. one can only diffuse knowledge or persuasion to someone who knows less than you or who has a less favourable attitude than you. Ideally, the individuals would be homophilous regarding other societal variables even though they are heterophilous regarding the innovation. Such heterophily manifests itself in the differences between innovators and early adopters or between early and late majorities.

■ **Summary**

It is without question that the diffusion of innovations in a social environment is a different proposition to the reconfiguration of the network-layer of a distributed group of nodes in a mobile ad hoc network. Nonetheless, the diffusion literature provides some well-rounded and well-reasoned descriptions of the characteristics of the diffusion process as it relates to the organisation and structure of some societies with regard to their adoption of innovations. The model is flexible as it is based on the actions of autonomous individuals who are influenced both by societal norms and a desire for individual advancement.

The diffusion model may be used to extend and adapt distributed agreement techniques for use in a mobile ad hoc network. Many of the elements of the innovation-decision process, and the descriptions and characteristics of the actors party to the various stages of the process, may be imposed on nodes in the mobile ad hoc network setting. An ad hoc network is not a society where individuals have free choice and can act in an irrational manner. The manner in which individuals learn of, form attitudes towards, and decide upon the innovations may be used to flesh out the design of a reconfiguration process which depends on strong local communication and weaker regional communication. Other aspects of the model such as the reaching of critical mass for interactive innovations and decisions based on *trial-by-others* are also applicable to the reconfiguration process.

7.5 The Reconfiguration Component

The reconfiguration component builds upon the network observation techniques of the type that were presented in Chapter 5 and the self-stabilising effects of the auto-configuration component that was presented in Chapter 6. The design of the reconfiguration component takes its cues from both the distributed consensus literature and the diffusion of innovations literature. The detail of the reconfiguration component will be laid out in the subsequent sections. The component is completely distributed, relying on opportunistic communication between nodes on both a local and regional basis.

The component operates in three stages; the Persuasion Stage, the Decision Stage and the Implementation Stage. The component's Reconfiguration-Decision Protocol (RDP) extends and modifies elements of the innovation-decision process for the task at hand. This section briefly introduces the three stages in the reconfiguration process before going on to discuss the background to the design of the component from a practical point-of-view. Three experiments are presented in Section 7.7 to demonstrate the protocol.

7.5.1 Overview of the Component's Protocol Design

The design of the RDP is influenced by the strict limits within which distributed consensus protocols may be implemented and it is also influenced by the flexibility exemplified by the innovation-decision processes. It has been shown that the making of decisions in an ad hoc network using distributed consensus techniques is non-trivial due to the challenging characteristics of the system. The dynamic nature of the network and the poor quality of communication channels, in comparison to typical distributed system settings, exacerbates the conditions which underlie the premise for the FLP impossibility result. However, local decision-making is possible, as exemplified by the local group membership scheme [Bri02] in which each node maintains a

correct local view; a local group membership is akin to a precise neighbourhood sensing scheme which higher service layers may rely upon to be accurate. While the *decision*, i.e. the agreement among the nodes of what constitutes the local view, does not appear to be a very complex one, it serves to demonstrate the difficulty of implementing an agreement protocol in an ad hoc network that meets the validity, integrity and termination conditions underlying the design of such a hard-state decision-making process.

Also, the MANETconf [Nes02] scheme modified the Ricart-Agrawala [Ric81] mutual-exclusion algorithm so that it was possible to implement that mutex scheme in an ad hoc network. The modification to the decision protocol involved the use of soft-state termination techniques. The MANETconf scheme also involves communication between a proposing node and every other node in the system; it makes use of multi-hop routing services. This soft-state decision-making approach allowed the authors to implement the scheme over a multi-hop network that could experience delays and partitioning.

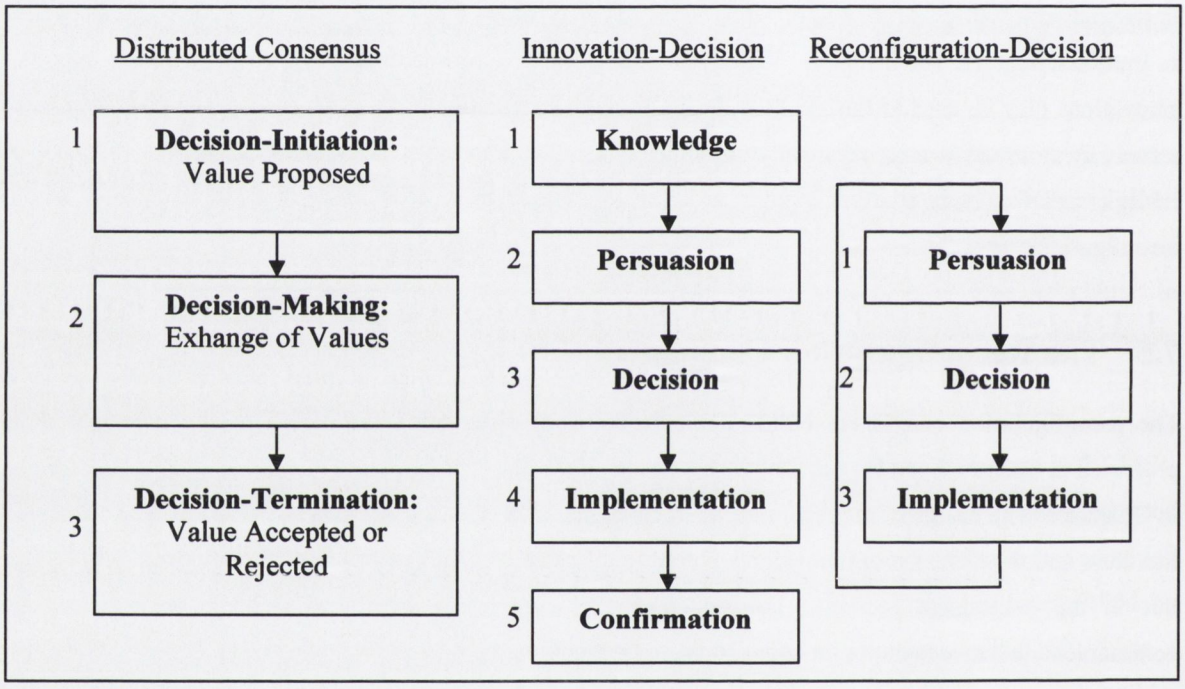


Figure 7.2 Three Decision Models

The RDP modifies the general layout of the innovation-decision process format, shown in Figure 7.2 alongside the three steps involved in a distributed consensus protocol. The distributed consensus model and the innovation-decision model are both triggered when a proposal is made to the system. In the distributed consensus model a decision variable is proposed. The next step generally involves consultation with every other node in the system concluding in a clear termination of the process. In the innovation-decision model, a system of individuals progress from first knowledge through to the persuasion, decision and implementation stages before seeking confirmation of their decision; the confirmation stage may lead to a replacement discontinuance if the decision-making unit is dissatisfied with the innovation. In contrast, the RDP jumps straight to the consultation stage, step two of the other approaches, without any proposal being made. The knowledge stage is not necessary in this system as the routing protocols are not true innovations. Each node is aware of the existence of each of the available protocols. Instead, the system may be seen as one in which

three protocols have been innovated and the nodes are continually seeking to confirm that their current choice is optimal for the given regional network conditions.

The Persuasion Stage of the RDP builds on these techniques that have demonstrated that a limited amount of distributed decision-making is possible in mobile wireless ad hoc networks. The Persuasion Stage enables each node to make decisions based on information received from neighbouring nodes. Even though the participants in the decision-making process are within one-hop of each other, i.e. they are within a bounded transmission time of each other, this system does not rely on the nodes entering into an active two-way dialogue as nodes are not allowed to originate their own signalling, i.e. they are not allowed to demand that the network-layer transmits a packet solely for them. A two-way dialogue involves a request/reply process which can stall a protocol if the reply is lost or delayed.

Instead, the Persuasion Stage involves a continuous process of passive observation and analysis of the networking conditions. Reconfiguration control beacons are piggybacked onto other outgoing packets which have been generated by a primary network-layer component. Each node makes soft-state decisions which are based only on passively received information and which may change as that information grows stale and expires. If the nodes manage to sustain a soft-state decision for a time, without altering it, then they enter the Decision Stage followed immediately by the Implementation Stage. The Implementation stage completes an execution of the reconfiguration-decision protocol and the nodes then re-enter the Persuasion Stage. The main elements of the RDP processes may be summarised as follows:

- Each node continually observes its networking conditions and shares that data with its neighbours.
- Each node repeatedly evaluates a soft-state preference based on its observations, the observations of its neighbours and its neighbours' soft-state decisions.
- A hard-decision is made when a group experience of the networking conditions is sustained.
- Decisions can be implemented using a *big-bang* or *creep* approach.

7.5.2 Regional Reconfiguration and the Auto-Configuration Component

Due to the impossibility of knowing what constitutes the 'entire' network in an ad hoc setting, the reconfiguration happens within a region of the network. A region describes a group of nodes that are sufficiently well connected to form an ad hoc network and to pass reconfiguration beacons between them. Note that the nodes do not have to be sufficiently well connected to form a multi-hop ad hoc network, i.e. data does not have to be forwarded by any node. The RDP only requires local connectivity, even if it is very unstable for the underlying ad hoc routing protocol. This lower demand on the network connectivity makes the RDP robust when the current routing protocol is struggling under the prevailing conditions and should be reconfigured. In a special case, a region also describes an isolated single node. A region will rarely, if ever, encompass the entire population of nodes in the network due to the inherent mobility of the nodes, the occurrence of node disconnections at the network edges and network partitioning.

The RDP is executed within a region or part thereof; the process does not extend beyond the region's boundaries. Referring back to the innovation of diffusions literature, and specifically with regard to the impact of the homophily of individuals, it was suggested that societal observations indicate that individuals

who speak the same language and share attributes are more effective at mutually understanding each other. In this system, and specifically with regard to the RDP, only nodes within a homogeneous region may partake in the reconfiguration process for two reasons which are analogous to the homophily argument. Firstly, nodes using the same protocol share a common means of communication; the rate at which nodes receive reconfiguration beacons from each other is dependent on the activity in the network. OLSR nodes will receive beacons at regular intervals whereas the receipt of beacons by DSR nodes will vary depending on the traffic loading on the network. Secondly, the networking observations collated by nodes are dependent on their routing protocol. Nodes operating the same protocols collate network measurements using the same signalling techniques. The fidelity of the observation information generated by nodes operating the same protocols is also similar, as evidenced in Chapter 5.

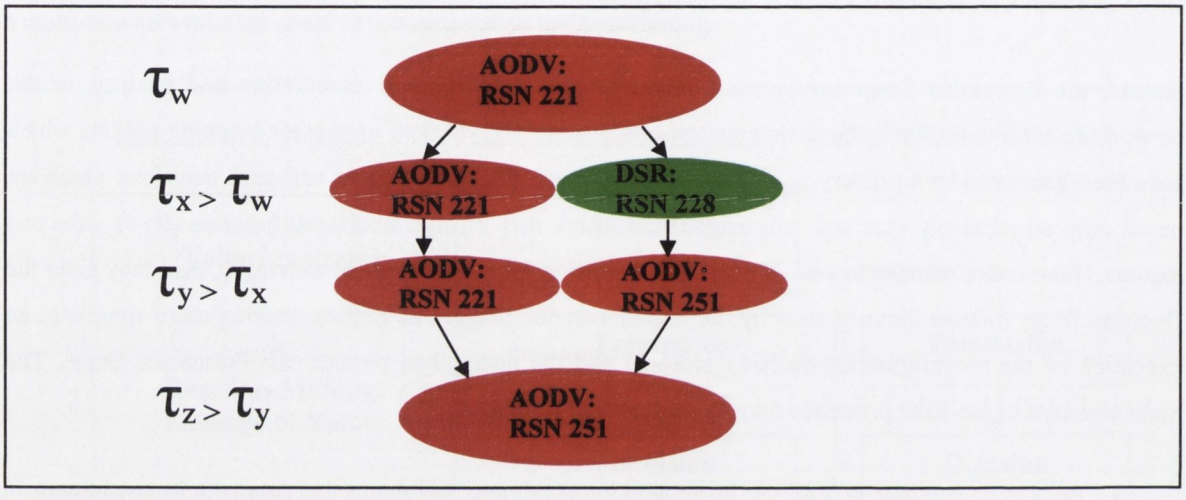


Figure 7.3 Regional Fragmentation

Furthermore, the reconfiguration component enables the region to fragment into separate regions if the networking conditions exhibit a spatial variation that demands such action; the reconfiguration component does not require the agreement of all nodes in the region in which it is operating. Such fragmented regions then conduct their own independent evaluation of the networking conditions but they may recombine at a later time. To explain, consider an AODV region which is depicted in Figure 7.3. The AODV region (Region Sequence Number (RSN): 221) is engaging the RDP and when it executes the Decision Stage the network fragments into two regions. These two regions now execute their own RDP processes. The Persuasion Stage soft-state decisions of the two regions do not influence each other. However, if the DSR region (RSN:228) decides at a later stage to change back to AODV again, then its RSN will go to 251. The auto-configuration CCDR protocol will ensure that the entire region adopts the higher RSN and the enlarged region will operate the RDP as one region again.

The flexible multi-protocol network-layer uses two configuration components: the reconfiguration component being described here and the auto-configuration component described in the last chapter. The reconfiguration component's decision protocol must be executed in a stable, homogeneous setting. The homogeneity requirement was motivated above and the stability refers to the ability of a region to correct the effects of naturally occurring, but disruptive, events such as nodes booting-up, joining, leaving and migrating at any time. The auto-configuration component ensures that such events are corrected as soon as a conflict is

detected. The auto-configuration component provides the network configuration stability required to execute a protocol in which decisions are not reactive and instantaneous, but based on sustained observation.

The policies of the auto-configuration protocols supersede those of the reconfiguration-decision protocol. The network, through the operation of the auto-configuration protocols, only permits nodes to form homogeneous regions. If a node's configuration has to be corrected by the auto-configuration protocol, then it abandons its current soft-state decisions and re-enters the Persuasion Stage using the observations generated since it was corrected. The only exception to this is when two regions operating the same protocol, but with different RSNs, come together, as depicted in Figure 7.3. In this scenario, the nodes correct their RSNs but continue to operate the RDP as normal as there is no substantive change to the network, i.e. the signalling mechanisms are the same in both of the merging regions. The Region Sequence Number is simply an administrative identifier which is only of use when a protocol conflict occurs.

7.5.3 Opportunistic Communication

The communication between nodes during the execution of the RDP and the connectedness of those nodes during the execution of the protocol are interdependent issues. The diffusion of innovations literature introduced the concept of mass media and interpersonal communications. Mass media channels are used to raise awareness of an idea; interpersonal networks are used to make arguments for and against that idea. This notion is extended into the reconfiguration-decision process in the form of reconfiguration beacons. These local beacons are opportunistically issued by each node to share soft-state decision information at the Persuasion Stage and to enforce a hard-state decision at the Decision Stage.

Like the persuasion stage of the innovation-decision process, and unlike that of a distributed consensus protocol, such as the Ricart-Agawala mutex protocol, the RDP Persuasion Stage builds on the 'interpersonal' communication channels that exist between the peer-adopters, i.e. the persuasion stage builds on the local exchange between neighbouring nodes. The RDP Persuasion Stage does not rely on multi-hop communication with non-neighbouring nodes. Any reliance on the explicit multi-hop unicasting services violates a core premise underlying the design of a robust reconfiguration protocol which cannot, of itself, be the client for which the multi-hop routing services are provided.

Reconfiguration beacons, which differ in intent depending on which stage the process is at, are configured and issued by the reconfiguration component according to the RDP's rules. As the reconfiguration component is a secondary network-layer component, such beacons are not actually broadcast by the node until the next time that a primary network-layer component issues a packet itself. In practical terms, this means that a node operating the OLSR protocol will not issue a reconfiguration beacon until the next time that it issues a control packet, such as a hello or topology-update packet, or a data packet. The reconfiguration beacon is simply queued pending the occurrence of such a primary network-layer component event. This process is referred to as opportunistic flooding as the reconfiguration component makes use of the existing out-going signalling to effect its business. A node's use of opportunistic communication, together with its passive observation of incoming ambient signals, ensures that this secondary component does not alter the various signalling profiles of the ad hoc network's regions.

7.6 Reconfiguration-Decision Protocol (RDP)

The Persuasion Stage of the RDP is the most important element of the protocol. It is during the execution of this stage of the protocol that each node decides which protocol suits *it*, based on the node's knowledge of its local network state. The node's motivation for making that decision varies depending on the local network state.

7.6.1 The Persuasion Stage

The persuasion stage of the innovation-decision process is concerned with nodes forming an opinion about an innovation; they are either in favour of the innovation or against it. This attitude does not always lead to the node adopting the innovation, but is a good indication of what it will do in the decision stage. Analogously, nodes in an ad hoc region are continuously engaged in forming an '*opinion*' about their preferred ad hoc routing protocol at all times. A node's preferred choice at any given time amounts to a soft-state decision; the node's hard-state decision is reflected in its current working routing protocol. A node's soft-state decision *may* be in agreement with its hard-state decision.

Individuals in the diffusion of innovations model make their decision based on their adopter classification which is a reflection of how fast they are to adopt a new idea. To recall, the classifications were Innovator, Early Adopter, Early Majority, Late Majority and Laggards. The need for such classifications is based on the individuality and heterogeneity of the decision-making units in a society; the decision-making units do not use uniform techniques to make their decisions. The classifications are designed to capture characteristics in the decision-making units with regard to their exposure to information and their willingness to adopt.

The diversity of the decision-making units extends to the ad hoc networking context with regard to the nodes involved in the RDP. Consequently, the notions of Early Adopter, Early Majority, Late Majority and Laggard Nodes are introduced into the RDP to define the decision models used by nodes in the Persuasion Stage. Nodes make their soft-state decisions based on a series of rules that relate to their classifications which are closely analogous to their innovation-decision counterparts. Each node changes the decision-making rules that it uses based on the network observations available to it and its knowledge of the decisions of its neighbours. While it is preferable to make a decision based on evidence alone, nodes are also enabled to make decisions that are influenced by the choices of their neighbours. For the purpose of clarity it should be noted that a node that is referred to as an Early Majority (EM) Node, for example, is an ad hoc node that has made its decision based on EM decision rules.

Early Adopter – EA – makes decisions based on observations only – requires the observations available to it to be somewhat homogeneous

Early Majority – EM – follows the EAs' decisions' as the observations available to it are not indicative of a particular option subject to a threshold

Late Majority – LM - peer pressure, based on the increased utility of agreeing to a change in a dependent system variable, persuades an LM node subject to a threshold

Laggard – LD – makes no decision as it is not sufficiently well-connected to the network

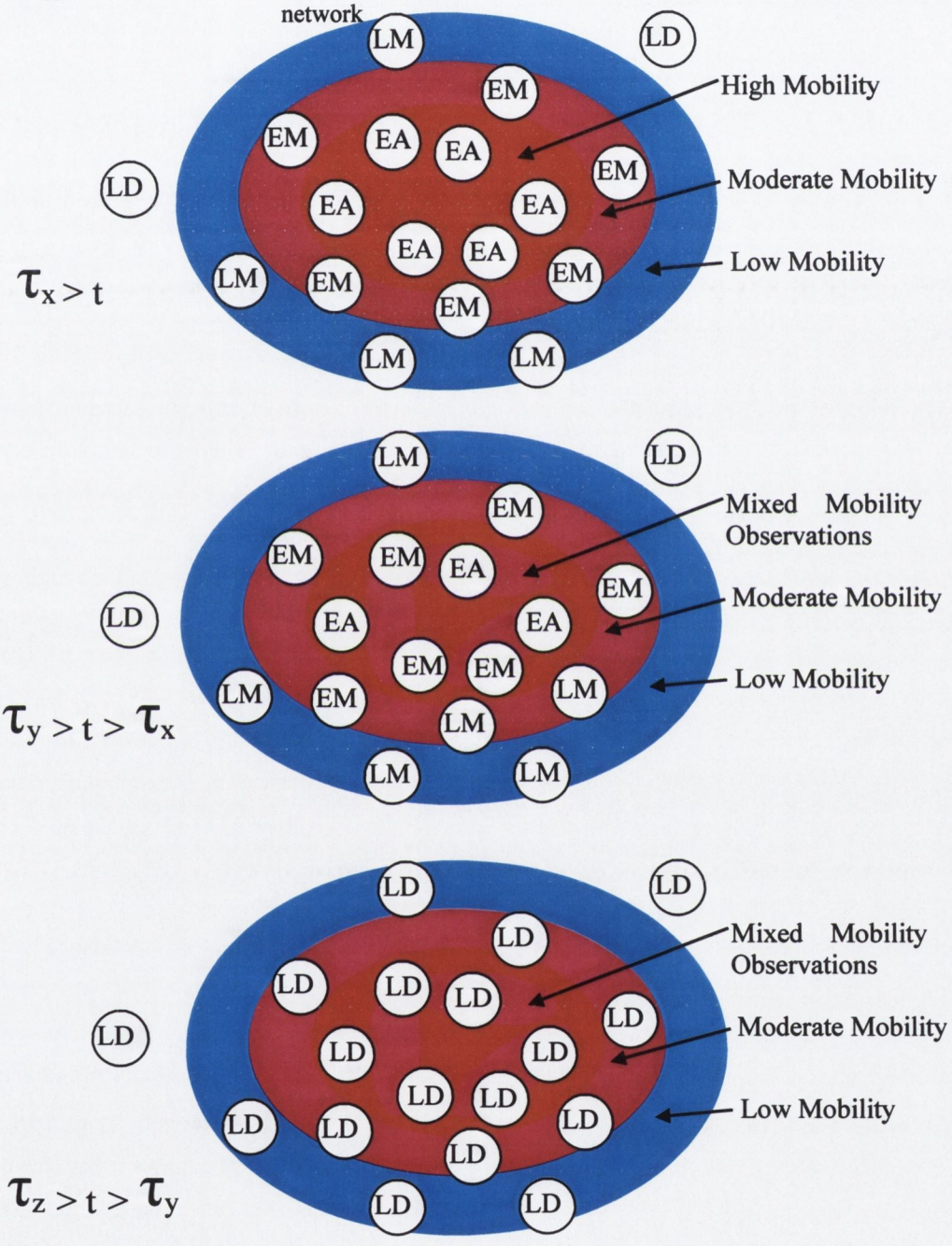


Figure 7.4 Early Adopters Have Similar Experiences

A simple scenario is illustrated in Figure 7.4 to illustrate the basic process of decision-making in the Persuasion Stage. The EA nodes at the centre of the network, for a time before τ_x, have been experiencing

similar conditions of high mobility (for arguments sake, say that mobility is the dominant factor in their choice of protocol) and they share their observations of these conditions. Each of these nodes is able to make an EA decision as its observations concur, within some bounds, with those of its neighbours. These EA nodes broadcast their decisions locally. Away from the centre of the network, the mobility conditions change and the average measurements indicate that the area is one characterised by moderate to low mobility. The nodes in this area of the network tend to have dissimilar network observations to their neighbours. However, the nodes in the moderately mobile area are aware that near-nodes are experiencing more consistent conditions as they have made, and announced, EA decisions to that effect. These nodes then make EM decisions in which the node is persuaded to make a decision on the strength of the EA nodes' decisions. Further out, the LM nodes have equally inconsistent observations of the network to those of their neighbours. However, unlike the EM nodes which are influenced by the strong EA decisions, the LM nodes do not have such neighbours. The LM node is swayed by the fact that its neighbours have made a soft-state decision which may affect its own state. As the innovation in this system is an interdependent one, i.e. it is of no using it unless other nodes do, the LM node will be isolated from the other nodes if it does not tow the line with regard to what a number of its neighbours have decided.

Both the EM node and the LM nodes' decisions are subject to the control of individual thresholds which dictate how easily persuaded a node can be. For instance an EM node may be required to have a majority of its neighbours as EA nodes before it adopts their decision whereas an LM node may have lower threshold that allows it to adopt the most popular decision that its neighbours have declared. The value to which the thresholds are set dictates how easy the persuasion process will spread. Setting the threshold too high would require a high degree of homogeneity within the network whereas lessening the threshold would allow the spread in a more heterogeneous environment.

Finally, some other nodes in the system are bound to find themselves with poor connections to the main network partition. These nodes collate observation data that is inconsistent with their neighbour's data and cannot form a decision using one of the above-mentioned models. Instead, these LD nodes make no decision. They will follow whatever decision is made by other nodes, if a decision is made.

As can been seen, the process is an iterative one in which the nodes with less confidence in their observations, due their inconsistency with their neighbours, are persuaded by the decisions of more confident nodes. These nodes in turn influence the decisions of still weaker nodes. However, it is important to note that the process of persuasion is not one-way. If the characteristics of the area of high mobility in the network depicted in Figure 7.4 were to change, some time after τ_x, such that some EA nodes became more static, then the nodes may collate conflicting observations reflecting the more heterogeneous nature of that area of the network. In such a scenario, the EA nodes would lose confidence in their decision. If the EA nodes cannot rely on their network observations, then they will move on to the next weaker decision model, that of the EM node. However, once the EA decisions are removed from the system, the persuasion that has built up in the area will break-down since the EM nodes will have to down-grade their decision and then the LM nodes will have to down-grade also. Within a few iterations of evaluating decisions and opportunistically broadcasting those decisions, by time τ_z, the network will have reduced to a collection of LD nodes which cannot make *any* decision. Such a conclusion would fit with the heterogeneity which is persisting in the network.

■ **Conflicting Objectives**

As evidenced in the network example above, when there is a conflict between nodes as regards their network observations, then it is not possible to make a decision; local agreement is a fundamental building block of this model. If the nodes cannot agree, they are left to use the protocol that they are already configured with. They may have been reconfigured to use this protocol at an earlier stage or their configuration may be the result of the auto-configuration component having corrected their states after some disruptive event.

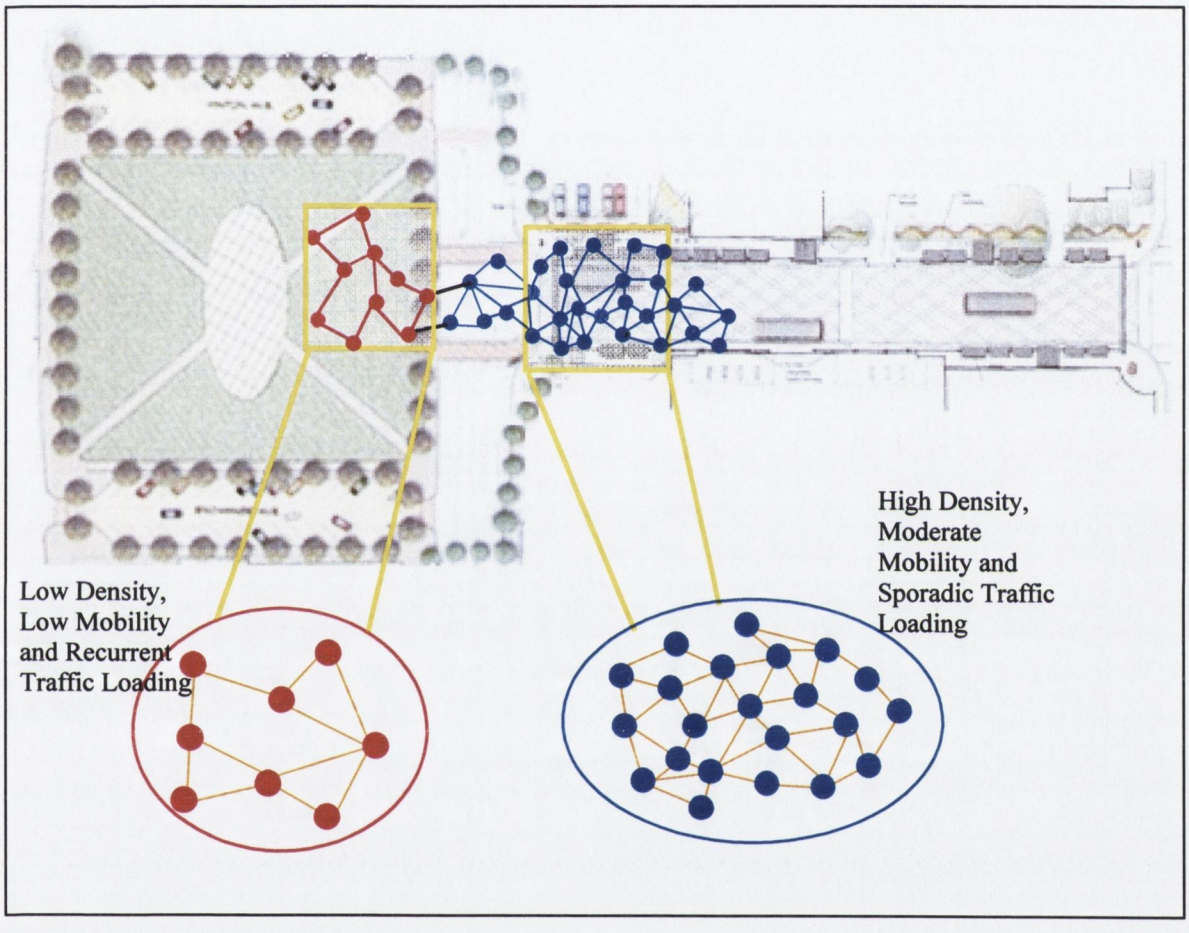


Figure 7.5 Urban Networking Scenario

Conflicting network experiences may also exist between nodes on a wider scale. The likelihood is that a real-life network, such as the network depicted in Figure 7.5, will exhibit different conditions in disparate areas of the network depending on the nodes' autonomous movements and the uses to which the nodes are put. But in any network, no matter how large and diverse in character it is, each node experiences the network on a local basis and these experiences may be measured by local, myopic metrics. While local measurements do *not* describe the larger network environment, they do describe characteristics of a particular space within that network at a given time. Referring to Figure 7.5 again, it may be seen that the network extends from an urban park out onto a street. While the region on the left is characterised by low mobility, low density and recurrent traffic loading, the area on the right is characterised by moderate mobility, high density and sporadic traffic loading. A decision model like that used in the RDP enables the network to configure itself with regard to its local and regional observations. The fact that the network is composed of

areas in which certain combinations of networking characteristics dominate means that the RDP's Persuasion process can run as far as the Decision and Implementation Stage.

7.6.2 Soft-State Decision Making

As noted in Chapter 6, each node sends out a Node Status Header in every packet that it issues. Node Status Headers have neighbourhood scope only and contain information regarding the issuing node's active protocol, Region Sequence Number and External State Vector (ESV) Data. Each node also issues a Reconfiguration Status Header in each packet along with the Node Status Header, shown in Figure 7.6. The beacon has two fields: the Decision, i.e. which protocol the issuing node prefers, and the Decision Type field which indicates which decision model the node used to arrive at the soft-state decision. The Decision Type field may have the following values; EA (Early Adopter), EM (Early Majority), LM (Late Majority) , LD (Laggard) and HS (Hard-State). The Laggard value indicates that the node could not make a decision and the Hard-State value is used to enforce a decision at the Decision Stage.

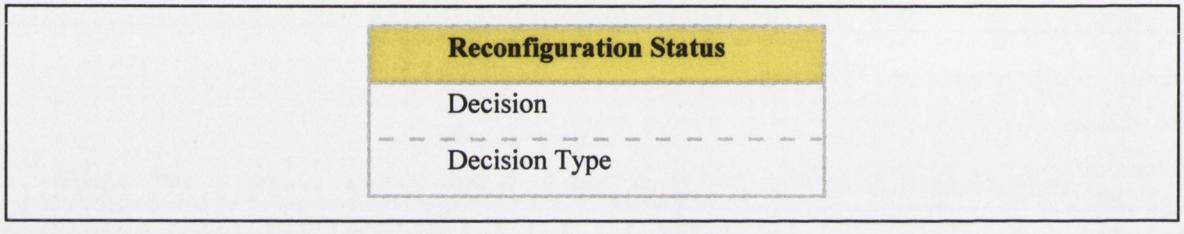


Figure 7.6 Reconfiguration Status Header

The Neighbourhood Table shown in Figure 7.7, which was introduced in Chapter 4 and discussed further in Chapter 6, is used by the reconfiguration component to record information received from neighbours regarding their ESV data and soft-state decision choices. Entries are added to the table, refreshed and removed in accordance with the techniques associated with the routing protocol in use.

Neighbourhood table

Entry	Active Protocol	Neighbour ID	Region Sequencer Number	ESV Data	Soft-State Decision/ Decision Type
0	OLSR	14	5	-	-
1	OLSR	17	5	-	-

Figure 7.7 Neighbourhood Table: ESV Data and Soft-State Decision

Each node periodically evaluates its recorded observations in order to arrive at a soft-state decision using rules based on the decision hierarchy which places the Early Adopter decision, which is based on network observations alone, at the top and the Laggard decision at the bottom, as illustrated in Figure 7.8. Each decision is based on the information that is available to the node when it executes an evaluation round. Initial Early Adopter decisions, i.e. the first decisions that can be made in the system, are made on the basis of a node's observation that its network observations match those of its neighbouring nodes, which leads them to have a common preferred choice of ad hoc routing protocol. The aim of the reconfiguration protocol is to

allow nodes to loosely exchange network observations and soft-state decisions that reflect the underlying heterogeneity or homogeneity of their experiences. If the networking conditions are such that neighbouring nodes do not have a common experience of the network, then those nodes will not progress the RDP towards a hard-state decision.

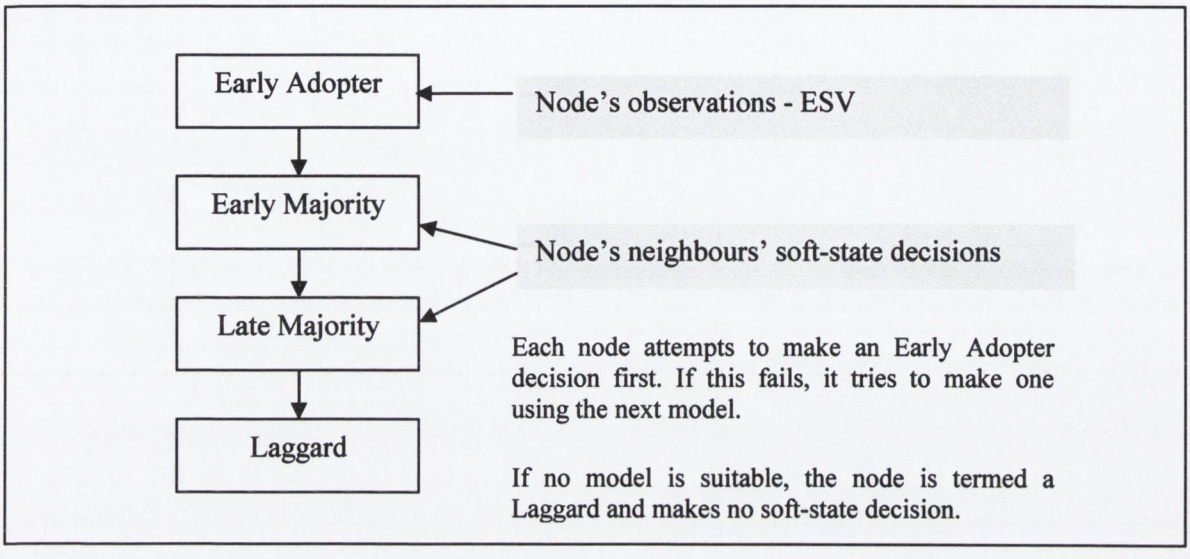


Figure 7.8 Decision Hierarchy

Each node evaluates its soft-state decision periodically at a short fixed interval. This period is locally variable; it specifies how often a node takes a snapshot of its local network state. A node that becomes isolated and steps down to a reactive protocol can suspend the Persuasion Stage and stop evaluating its decision or set a larger value for the timeout. An isolated node then re-enters the Persuasion stage if it reconnects to the network.

■ Flushing Stale Data

It is evident that the accuracy of a node's Neighbourhood Table varies from protocol to protocol, as demonstrated in Chapter 5. A neighbour will only persist in an OLSR cache for a short time before it is expunged. OLSR nodes are only allowed to miss two Hello messages from a node before they remove the entry, whereas a node can be stored in a DSR cache for up 300s before the node expunges it. The difference occurs due to the fact that OLSR is an active link-state sensing protocol and DSR is a *lazy* reactive protocol; DSR does not actively check the status of a cached neighbour unless it tries to use the out-going link and it fails. Consequently, DSR nodes in more mobile environments tend to have neighbour entries which are no longer valid from a routing point-of-view. However, the RDP is not concerned with the validity of the information as such; it will not be attempting to make use of the out-going links. The RDP is only concerned with the associated neighbour entry information such as the ESV Data and soft-state-decision data, to provide it with a view of the local network's conditions as seen by neighbouring nodes. If the information persists in the Neighbourhood Table due to the routing protocol's timeout mechanisms, the RDP may use very stale observation data if it uses every entry in the Neighbourhood Table. To the RDP, neighbouring nodes are simply a timed series of data sources, nothing else. To combat this problem, the decision-making node can employ a suitable filter, e.g. an EWMA filter, to place greater weight on the most recently received data and to diminish the effects of data that has not been refreshed.

■ **Early Adopter Nodes**

In an innovation-decision process, early adopters should make judicious choices which are based on sound reasoning. Innovation-decision early adopters lead other individuals with less knowledge or experience on which to base their decisions. In this system, the Early Adopter Node is a node which follows that model by basing its decision purely on the evidence that it collates from the networking environment, i.e. its own observations, and possibly its neighbours' reported observations. The Early Adopter Node also maintains a Decision Counter which is incremented each time the node reconfirms its Early Adopter soft-state decision. The counter is set to zero if the node has to down-grade its decision.

Figure 7.9a depicts a decision-making node, Node x, at time τ_y, just as the node is about to evaluate its decision. Node x has 8 one-hop neighbouring nodes from which it has received reconfiguration beacons containing the nodes' ESV data sets.

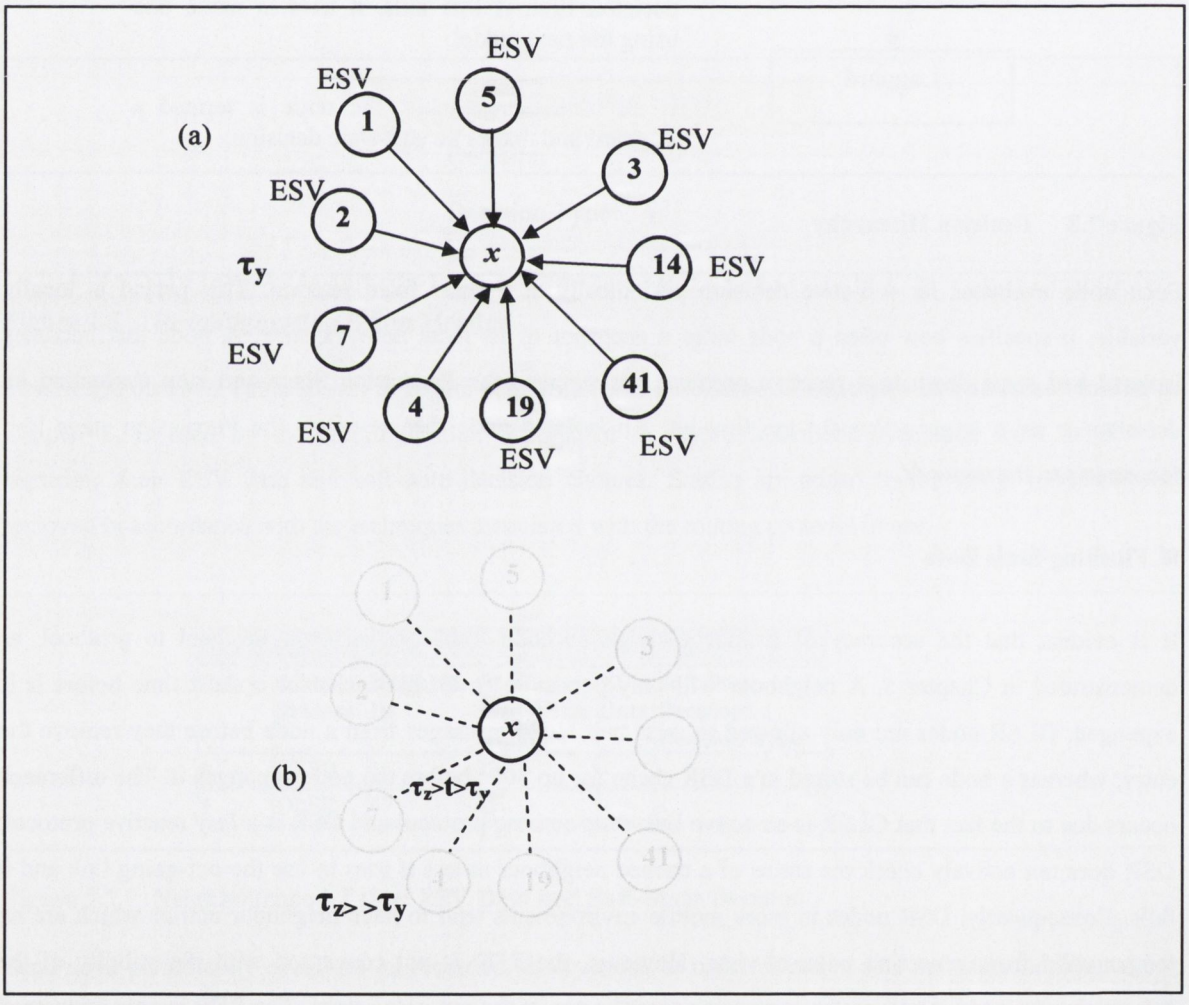


Figure 7-9 Early Adopter Decision: Node's and Neighbours' Observations Only

Each node examines the ESVs from its neighbours according to the control imposed by a suitable filter which discriminates in favour of the most recently cached beacons. It is important to note that Node Status and Reconfiguration Headers are *received by* nodes rather than being *sent to* nodes. The distinction is that this protocol, and indeed the auto-configuration protocols, only rely on information that a node passively receives

from neighbouring nodes in the course of their normal signalling. The fact that Node 2 is a neighbour of Node x at this time does not imply that Node x is a neighbour of Node 2, i.e. the link cannot be assumed to be symmetrical. The configuration protocols do not attempt to sense the quality of the links either, as the OLSR protocol does, as this amounts to active neighbourhood discovery. The decision-making node, Node x, is only concerned with the nature of the prevailing networking conditions as observed by its neighbours and passed on to it through the reconfiguration beacons.

The decision-making node's External State Vector, i.e. its myopic observations, maps to a particular ad hoc routing protocol choice as was demonstrated in Chapter 5. However, the Early Adopter node does not make its decision in isolation from its neighbours. Unilateral choice is not possible as the decision affects other nodes. Instead, the node compares its own observations with the averaged observations of the data it has received. The node can only adopt an Early Adopter decision if its myopic observations are, within some threshold, closely matched to the averaged observations of its neighbours.

The Early Adopter model could use more sophisticated means to combine the observations that it has collated and to determine its soft-state choice. However, the manner in which the Early Adopter makes its decision does not affect the rest of the operation of the protocol. In that sense, the RDP represents a framework within which more sensitive observation collation and filtering methods could be designed.

If the node manages to make an Early Adopter decision, the node updates its Decision and Decision Type variables to reflect the node's current soft-state decision. The node beacons these values in headers of every subsequent packet that its routing protocol issues until such time as the node changes its Decision.

However, the case may arise where a node does not originate any traffic after it has made its decision or the node could be experiencing out-going asymmetry on each of its links, a fact that it is not aware of. In either case, the node's current decision will not be relayed to its neighbours and this decision will not have any further influence on the persuasion of other nodes. This turn of events, as depicted in Figure 7.9b does not affect the protocol as it does not depend on those other nodes receiving a confirmation beacon from Node x. The fact that other nodes would not receive Node x's decision does not mean that they are denied a confirmation of their decision, such is the nature of the soft-state approach to the design of this protocol.

■ Early Majority Nodes

The next category of decision is the Early Majority decision. This is a decision in which a node cannot make an Early Adopter decision, i.e. one based only on its observations. This situation occurs in ad hoc network as the experience of nodes may vary such that nodes do not collate a sufficiently similar network view. To counter the Early Majority Node's uncertainty about the data available to it, the node uses its knowledge of the Early Adopter decisions of other neighbouring nodes to influence its decision.

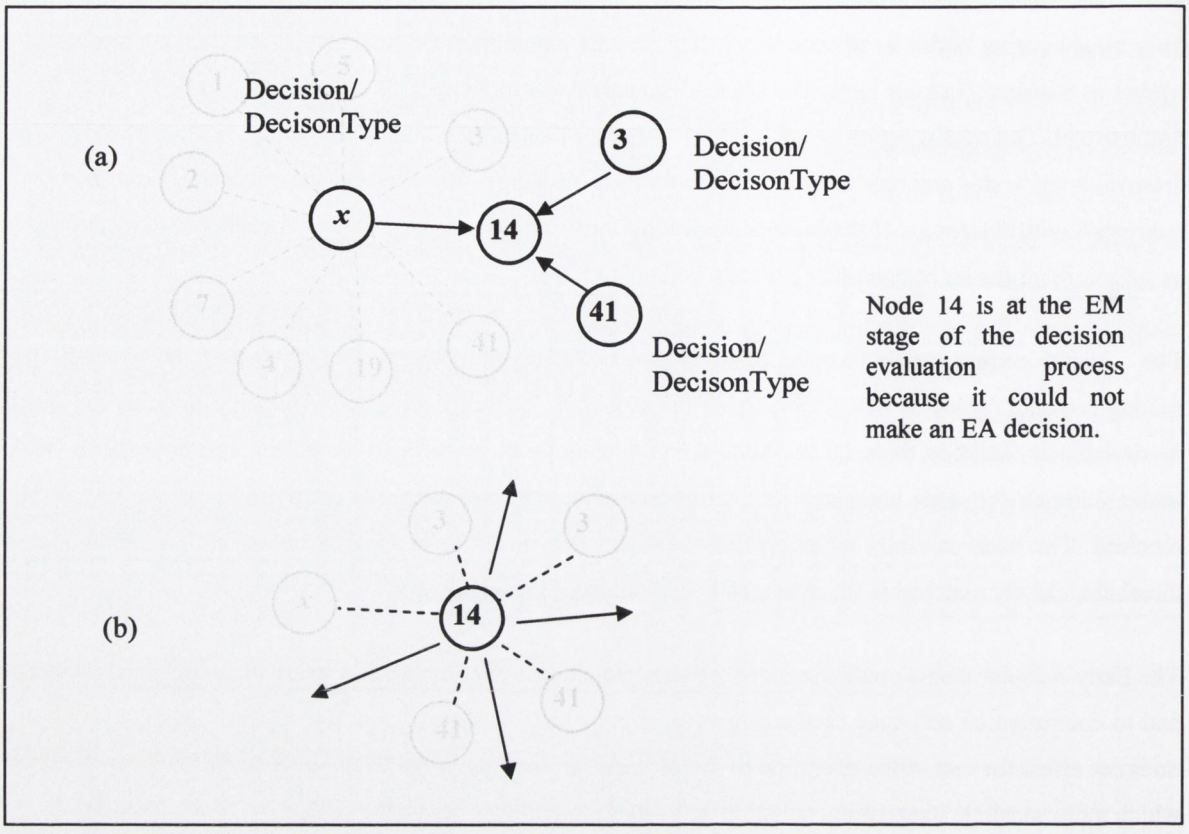


Figure 7-10 Early Majority Decision: Early Adopter Neighbours Only

Specifically, an Early Majority node adopts the decision reported by its neighbours if more than some threshold of them are advocating an EA decision for a single protocol. Consider Node 14 in Figure 7.10 which has received three beacons that have not been timed out by its route cache. One of those beacons came from Node *x*, which has adopted an EA decision. Depending on the Early Majority threshold value, the receipt of one EA decision may be enough for it to adopt the chosen protocol or it may demand more stability in the network by demanding that at least a simple majority of its neighbours have made an EA decision in favour of one particular protocol.

■ Late Majority and Laggard Nodes

The last category of decision making is the Late Majority decision. As the ad hoc routing protocol is an interactive innovation, i.e. its utility increases as the group of adopters grows, Late Majority Nodes are forced to adopt the decision of their neighbours if their observations are such that they cannot make a stronger decision, i.e. an Early Adopter or Early Majority one. Like the Early Majority, the *majority* in Late Majority does not refer to the threshold it must pass as an individual node before is can adopt a decision, but rather to its place as part of the later group of nodes that adopt a decision with much more ease once a core group of support, built on solid observations, has been founded.

The Late Majority Node is influenced by the number of its neighbours that have declared as EA or EM nodes for a particular protocol. The threshold used by the Late Majority Node is lower than that used by the Early Majority Node as it is more concerned with being associated with the most dominant protocol should that

protocol be adopted in the near future. If a node is unable to make a soft-state decision using one of the other rules, then it records its soft-state decision as Laggard

7.6.3 The Decision and Implementation Stages

The RDP progresses to the Decision Stage when one of the Early Adopter's Decision Counters reaches a level that indicates that the EA node has been advocating an EA decision for a defined period of time. Remember that the EA node increments its Decision Counter by one every time that it reconfirms its decision. By choosing appropriate values for the period in which nodes evaluate decisions and then choosing an appropriate number for the Decision counter the RDP can be allowed to progress rapidly through the persuasion stage or it can be made to move slowly through towards the Decision Stage.

The Early Adopter Node issues a Reconfiguration Status Beacon with the Hard-State in the Decision field. Each receiving neighouring node that has made the same soft-state decision, regardless of how it made the decision, adopts the hard-state decision by changing its protocol to the ad hoc routing protocol over to the new routing protocol and incrementing its Region Sequence Number to the appropriate value.

Those nodes then continue to beacon that hard-state decision for a set time. A hard-state decision cannot be down-graded like the other decisions. From then on, those nodes that have reconfigured their protocols will issue Node Status Header with an increased Region Sequence Number. As these nodes are connected, (they must be to have received the Reconfiguration Status Header), they attain a superior place in the auto-configuration CCDR protocol hierarchy. Consequently, neighbouring nodes that receive a Node Statue Header indicating the new header value must adopt the decision.

The node that issued the beacon does not change itself until it receives a beacon from another node indicating that the decision has been implemented by another connected node. This relates back to the point made with regard to Node *x* in Figure 7-8b.

7.6.4 Creep: Trial-By-Others

In a special decision-making case, nodes can execute the decision-making process unilaterally without reaching the standard that EA nodes must reach. These nodes which take a trial-by-others approach are influenced by neighbouring nodes that are already using the protocol which the decagon-making node has adopted as its soft-state decision. This situation occurs if the decision-making node is adjacent to another region, i.e. a group of nodes that are not like-configured. At least one of the decision-making node's neighbours will be operating in dual-mode. i.e. as a Boundary Node which is set up when two heterogeneously configured regions meet or when one region decomposes using this component.

In the network depicted in Figure 7-11, Node 2 has made a soft-state decision, of any kind, in favour of DSR. As Node 13 sits on a DSR/AODV region-to-region boundary, it has a DSR neighbour to which it is connected. If Nodes 2 and 13 were not connected then Node 13 would not be operating as a Boundary Node.

Node 2 operates a Decision Counter similar in intent to the EA's Decision counter. If it passes a preset level, the node may unilaterally enter the Decision Stage. In doing so, Node 2 does not send out a Hard-State Reconfiguration Beacon.

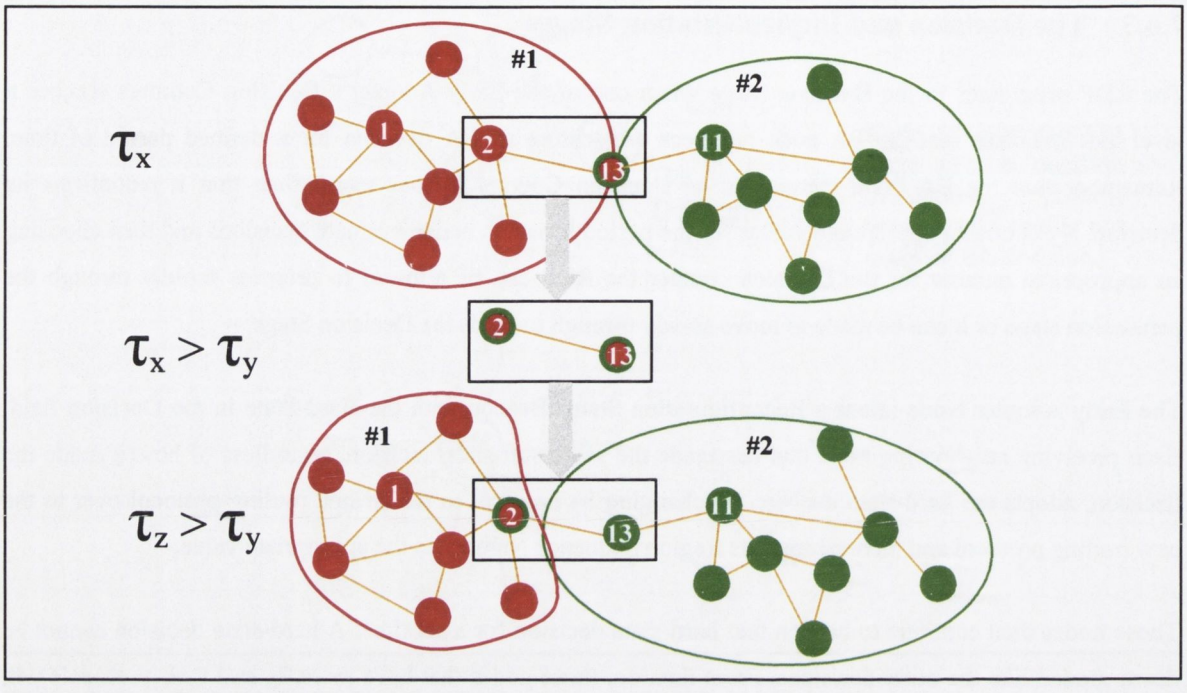


Figure 7-11 Trial-By-Others: Creep

When Node 2 executes this type of decision, which does not spread like a regular decision, it must take account of its neighbours. If the reconfiguring node's only neighbour is the Boundary Node which is influencing it, then the node changes over to the adopted protocol in a step-change. However, if the Node has other AODV neighbours in region #1, it switches to AODV/DSR Boundary Node status to maintain connectivity to the nodes in its first region. This is shown in Figure 7-11, where Node 2 sets up as a Boundary Node at time τ_y. The auto-configuration component then corrects the network as shown in Figure 7-11 at time τ_y. The Boundary Node, as exemplified by Node 13 at time τ_x , does not engage in trial-by-others approach as it is acting as a conduit between the two heterogeneously configured regions. The use of a trial-by-others approach along region-to-region boundaries lets the configuration of one region creep into another region as the conditions dictate.

7.7 Demonstrative Experiments

In this section, three example scenarios are presented, each of which demonstrates the feasibility of the RDP. These simulated scenarios also give some indication of the way in which the RDP performs. A suitable metric for the evaluation of such a protocol is the Network Recovery Time [Sub99]. Network Recovery Time is the time taken for a network to recover after a condition indicates the reorganisation of the network. Specifically, it is the time taken for the network, or a partition thereof, to become functional again after changes in traffic, mobility or other network characteristics. In the following experiments the Network Recovery Time is taken to be the time between the initial perturbation of the networking conditions and the

final reconfiguration of the last network node to the new protocol choice. The experiments were conducted under the same conditions as those experiments described in Chapter 5.

The RDP protocol was configured such that each node evaluated its decision every 1000ms. An EA decision had to persist for 10 iterations (i.e. the Decision Counter must reach 10) before the node could engage the Decision Stage. The choice of the length of the decision evaluation period and the Decision Counter value was based on some initial runs. In practice, the choice of such values would be subject to rigorous testing and evaluation to find the most suitable values. It is also possible that such values could be configured by each node on an individual basis in response to the observed local dynamics. For example, a node in a particularly volatile setting might insist that the Decision Counter value be set higher so that the node was sure as to the persistence of the change in conditions.

7.7.1 Changing Mobility Conditions

Scenario One, the 'changing node mobility' scenario demonstrates the ability of nodes to react to pertinent changes in the node mobility characteristics of a network. In this scenario, the mobility of a group of nodes is changed from that of a moderate mobility model to that of a high mobility model, after a time. This change has the effect of decreasing the average link duration experienced by each node as nodes create and lose links more frequently. The nodes move under a moderate node degree model and under a sporadic offered traffic loading model for the duration of the experiment. The details of such models are set out in Chapter 5.

Specifically, a group of 30 nodes were initially distributed in a space of 350m by 350m according to the random waypoint (RWP) mobility model [Bro98]. The radio transmission radii for each node were set at 100m. These parameters yield an average node degree of 8 neighbours per node, a moderate node degree. Initially, a moderate mobility model was used in this scenario; the RWP parameters were; minimum velocity -1m/s ; maximum velocity – 5 m/s and a pausetime of 20s. After a runtime of 400s, noted as τ_x in Figure 7.11, the mobility model was changed to represent a higher mobility model. Specifically, the RWP parameters which govern the movement of each node were changed as follows: minimum velocity – 5 m/s; maximum velocity – 10 m/s and pausetime 10s. For the purposes of this experiment the nodes were initially configured to use the OLSR protocol, the best protocol for the initial conditions.

To demonstrate the workings of the protocol, the experience of a single sample node in this experiment is described. The following points, which bear reference to Figure 7.12, provide a description of the pertinent aspects of the reconfiguration process for this node. Point **A** in Figure 7.12 draws attention to the two threshold levels which have been identified as crossover points, i.e. where changes in the node mobility are significant enough to warrant the use of another protocol. At this point, the sample node is experiencing conditions which mirror those of its neighbours, as is evidenced by the ESV data.

At Point **B** it may be observed that the node fails to make an EA decision but oscillates between an EM decision and an LM decision. At this point, it may be seen that there is a disagreement between the node's observations and the reported ESV observations. While the ESV observations are still in the range of observations for a moderate mobility model, the node's observations, which are due to its particular

interaction with the network, corresponds to a lower mobility scenario, hence the higher average link duration.

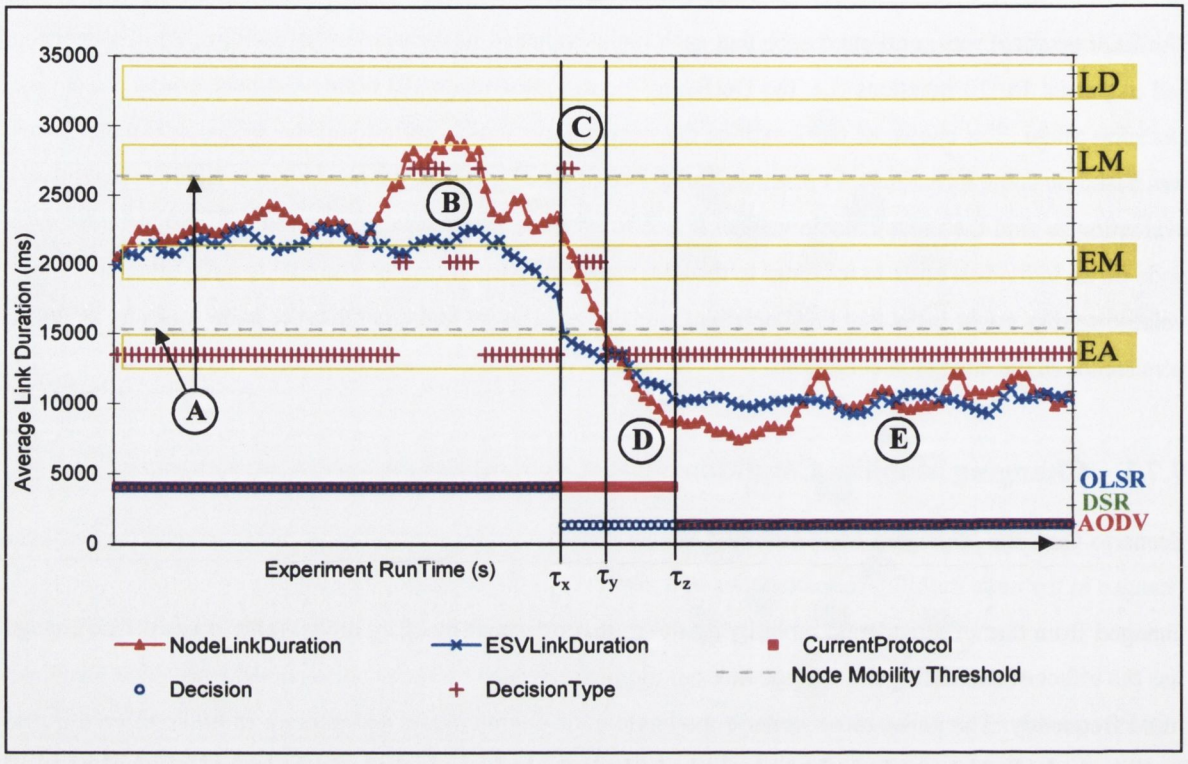


Figure 7-12 Reconfiguration in Response to Changing Node Mobility

Moving on to Point **C**, which occurs after τ_x, the mobility model transition point, it may be observed that both the node's observations and the reported ESV observations reflect the increase in the relative mobility of the nodes as the average link duration falls off. During this time, between τ_x and τ_y, the ESV beacons reflect the new state of the network faster that the node's own observations do. This may be a reflection of the fact that the node has been experiencing a localised maximum of lower node mobility as reflected in the extended link duration averages. Regardless, the ESV beacons lead the node to adopt an EM decision in favour of a new protocol, AODV.

By τ_y, both the node's observations and the ESV observations have fallen below the high/moderate mobility threshold. As the observations are now below an average of 15,000ms link duration, both observations indicate that the node/network is subject to higher mobility. At this point it can be seen that the node has changed its Decision, opting to select AODV as its preferred routing protocol by means of an EA decision. After successfully holding with this decision for 10 iterations, this node then progressed to the Decision Stage of the RDP, so that by τ_z the node has reconfigured itself to use AODV.

The Network Recovery Time for this group of nodes was 26s, i.e. the time between τ_x and τ_z for the whole group of nodes. The group time consisted of an average of 11s for each node to allow its observations to reflect the altered networking conditions, coupled with an additional average of 15s for the nodes to make and adopt the decision. The second time of 15s was due to some nodes' observations initially oscillating

above the threshold and thus resetting the Decision Counter to zero. This time also incorporates the time taken for the Decision and Implementation Stages to be enacted at nodes which were distant from EA nodes and consequently had to await the opportunistically broadcast hard-state decision beacons. The time taken for the sample node was 16s, 6s for its observations to reflect the mobility changes and then 10s for the decision to be made. The difference between the sample node and group times may be accounted for by the fact that clusters of EA nodes may form in different areas of the network in a shorter time when neighbouring node's observations converge quickly. In this case, the sample node's neighbourhood converged faster than the group observations did.

7.7.2 Changing Node Degree Conditions

Scenario Two, the 'changing node degree' scenario demonstrates the ability of nodes to react to pertinent changes in the node degree characteristics of a network. In this scenario, a group of nodes move from occupying a small space to occupying a larger space after a time. This change has the effect of decreasing the average node degree of each node as the nodes occupy a larger space. The nodes move under a low mobility model and under a recurrent offered traffic loading model for the duration of the experiment. The details of such models are set out in Chapter 5. In this scenario a group of 30 nodes were initially distributed in a space of 350m by 350m according to the random waypoint (RWP) mobility model [Bro98]. The radio transmission radii for each node are set at 100m. These parameters yield an average node degree of 8 neighbours per node, a moderate node degree. After the a runtime of 400s, the space within which the nodes may move was changed to 550m by 550m yielding an average node degree of 4 nodes per neighbour. For the purposes of this experiment the nodes were initially configured to use the OLSR protocol, the best protocol under the initial conditions.

To demonstrate the workings of the protocol, the experience of a single sample node in this experiment is described. The following points, which bear reference to Figure 7.13, provide a description of the pertinent aspects of the reconfiguration process for this node. Point **A** in Figure 7.13 draws attention to the two threshold levels which have been identified as crossover points, i.e. where changes in the node degree are significant enough to warrant the use of another protocol.

At Point **B** it may be observed that this particular node falls into a local maximum with regard to the number of neighbours it meets, causing it believe itself to be in an area of high node degree. At this point, it may be seen that there is a disagreement between the node's observations and the reported ESV observations. The ESV observations are still in the range of observations for a moderate node degree model. Again, as with sample node shown in the last scenario, the fact that this node cannot make any stronger decision than an EM decision indicates that the ESV beacons that the node has received indicate that the majority of other nodes have a different network view and have consequently made a conflicting decision.

Moving on to Point **C**, which occurs after τ_x, the node degree model transition point, it may be observed that both the node's observations and the reported ESV observations reflect the decrease in the average node degree as the space within which the nodes roam has increased. Just before τ_y, the ESV observations dip below the low/moderate node degree threshold and the node adopts both EM and LM decisions for DSR as

its observations conflict with its neighbours. However, by τ_y both the node's observations and the ESV observations have fallen below the threshold. At this point it can be seen that the node has made an EA decision in favour of DSR. After successfully holding with this decision for 10 iterations, this node then progressed to the Decision Stage of the RDP, so that by τ_z the node has reconfigured itself to use DSR.

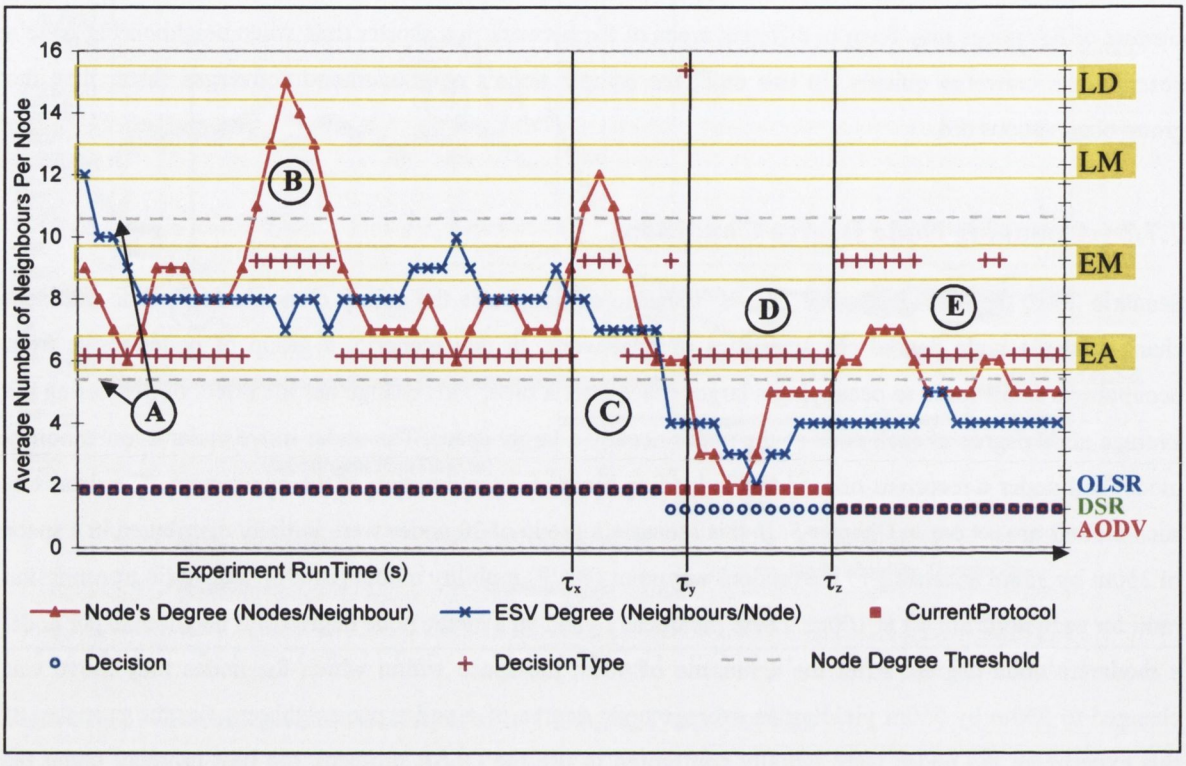


Figure 7-13 Reconfiguration in Response to Changing Node Degree

The Network Recovery Time for this group of nodes was 24s. This time consisted of an average time of 11s for each node's observations to reflect the change in node degree conditions. After this, the nodes took an additional 13s to enact a decision. For the sample node in this experiment, the recovery time, i.e. the time between τ_x and τ_z, was 18s.

Finally, at Point **E**, it can be seen the node's observations waver as it enters localised spots of the network that raise its perception of the average node degree. As the node's observations disagree with the ESV observations received by it, the strength of its decision oscillates between an EA and an EM in favour of its current protocol choice.

7.7.3 Changing Traffic Conditions

Scenario Three, the 'changing traffic' scenario demonstrates the ability of nodes to react to pertinent changes in the offered traffic load characteristics of a network. In this scenario, the type of traffic load offered to a group of nodes is changed from a recurrent traffic load to a sporadic traffic load after a time. The nodes move under a moderate mobility model and a high node degree model for the duration of the experiment. The details of such models are set out in Chapter 5. In this scenario, a group of 30 nodes were initially distributed in a space of 260m by 260m according to the random waypoint (RWP) mobility model [Bro98]. The radio

transmission radii for each node were set at 100m. These parameters yield an average node degree of 14 neighbours per node, a high node degree. Initially, each node in the experiment was offered a recurrent traffic load, i.e. each node was tasked with delivering data to a specific sink node, such as may occur in a sensor network scenario. After a runtime of 400s, the offered traffic load model was changed to that of a sporadic load. Each node was tasked with delivering data to a variety of nodes at different times and for different durations, as described in Chapter 5. For the purposes of this experiment the nodes were initially configured to use the AODV protocol, the protocol best suited to the initial conditions.

To demonstrate the workings of the protocol, the experience of a single sample node in this experiment is described. The following points, which bear reference to Figure 7.14, provide a description of the pertinent aspects of the reconfiguration process for this node. Point **A** in Figure 7.14 draws attention to the threshold level which has been identified a crossover point, i.e. where change in the characteristics of the offered traffic loading are significant enough to warrant the use of another protocol.

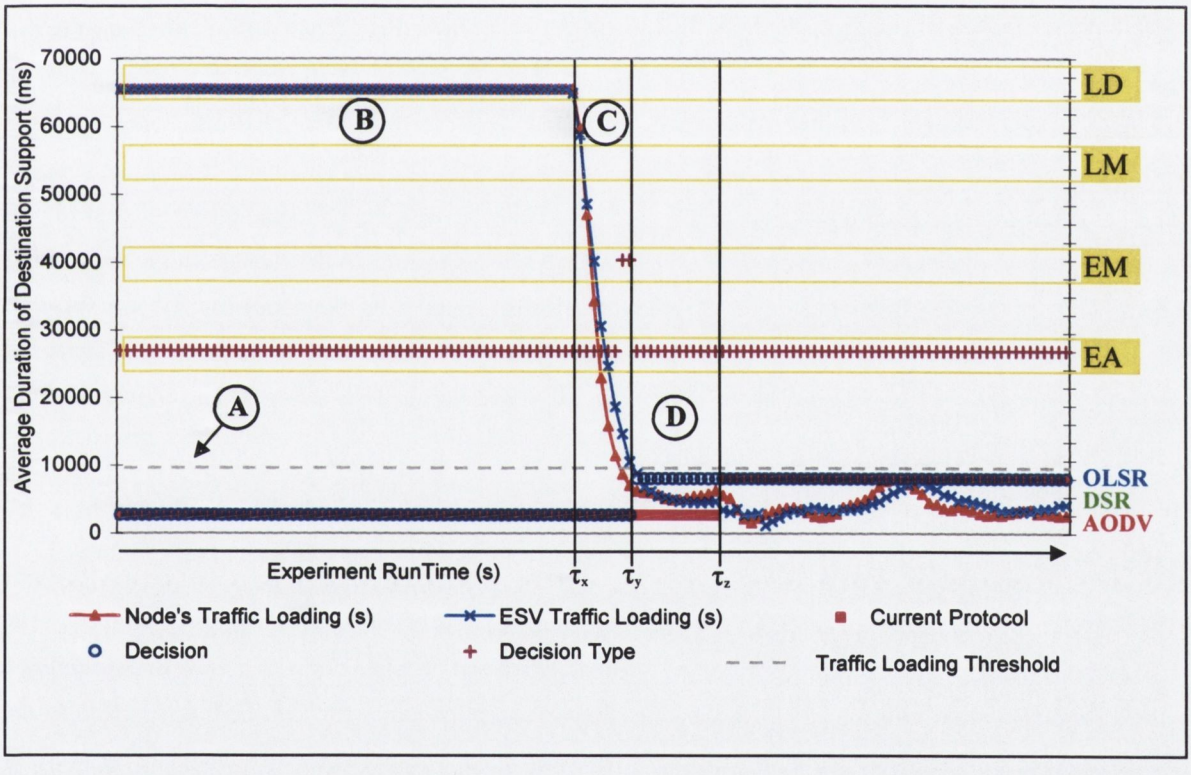


Figure 7-14 Reconfiguration in Response to Changing Offered Traffic Loading

At Point **B** it may be observed that both the node's observations and the ESV observations have plateaued. This has occurred as all nodes in the experiment are subject to recurrent traffic which means the average duration of support for a destination continues to grow as the experiment progresses. However, as the crossover point for traffic loading has been found to be at 10,000ms, the ESV beacons only use 2 bytes (maximum integer value - 65,356) to represent this metric. The EWMA filters used to record the observed data capture the actual average destination support times, but since this data is not shared by nodes in ESV beacons, the truncated data, upon which decisions are made, is shown.

Moving on to Point **C**, which occurs after τ_x, the traffic loading model transition point, it may be observed that both the node's observations and the reported ESV observations reflect the change in offered traffic load. It may be seen in the diagram, just before τ_y, that the node's observations dip below the threshold while the ESV observations are still above it. This leads the node to adopt weaker EM and LM decisions in favour of AODV, indicating that its confidence in that protocol is weakening. By τ_y, both the node's observations and the ESV observations have fallen below the recurrent/sporadic threshold. As the observations are now below 10,000s average destination support time level, the observations indicate that the node/network is subject to a more sporadic traffic loading. At this point it can be seen that the node has changed its Decision, opting to select OLSR as its preferred routing protocol. After successfully holding with this decision for 10 iterations, this node then progressed to the Decision Stage of the RDP, so that by τ_z the node has reconfigured itself to use OLSR.

The Network Recovery Time for this group of node was 19s. This shorter Network Recovery Time may be accounted for by the fact that traffic measurements are based solely on protocol requests made directly to the node's routing protocol. The metric used, the average destination support time, will reflect any change in the traffic loading on the node immediately as the requests for support for a destination will either cease or continue. This is in contrast to the radar-like mobility and degree metrics. The Network Recovery Time consisted of an average of 8s for each node before its observations indicated the need for change coupled with an additional average of 11s for the nodes to reach a decision. The sample node, represented in Figure 7.13, took 19s. For the three examples, the time taken for the nodes to react to the changes in the networking conditions relates to the filters in use. In all three scenarios presented here, the nodes use EWMA flip-flop filters with a gain value of 0.1 and moving range window of size 16. If the size of the window was increased or if the gain value was decreased then the filter would be slower to track the changes. The converse would also hold true.

7.8 Summary

This chapter has focussed on enabling the reconfiguration of the network-layer's routing protocol. The challenges and objectives for such a component were identified and thrashed out with a view to developing a robust reconfiguration component. Following the design blueprint that was identified in Chapter 4, the various elements of the reconfiguration component were identified. These are based on the recurring need for observation, analysis and decision-making. As the component is primarily concerned with the distributed organisation of the network based on the node's observation of their networking conditions, a review of the area of distributed consensus protocols was presented. However, this review demonstrated that the there are severe limits to the kind of system in which a hard decision protocol of the kind exemplified by traditional distributed consensus protocols can be implemented. However, some of the literature revealed that the use of soft-state decisions enables nodes to engage in decision protocols that do not require active confirmation techniques to be employed, a key hurdle in ad hoc network.

Consequently, a review of decision models used in other disciplines, which would be amenable to the conditions experienced in ad hoc networks, was undertaken. After investigating the theory of the diffusion of

innovations, it was found possible to construct a decision-making model which caters to the varying characteristics of the participants in a mobile ad hoc networking system.

A Reconfiguration-Decision Protocol was developed which is largely based on the theory behind the use of persuasion in the innovation-decision model. It was shown in this chapter that such a model is suited to the characteristics of an ad hoc network which cannot guarantee reliable communication between nodes or the constant presence of all the participants. Furthermore, the RDP was designed in compliance with the underlying assumptions, namely that it should not interfere with the signalling of the primary network-layer components.

8 Enabling Ad Hoc Internetworking

8.1 Introduction

As outlined in Chapter 3, one of the key requirements for a multimodal ad hoc network is the ability to internetwork between heterogeneously configured regions of nodes. This level of networking flexibility is necessary in a system that allows nodes to be configured to use one of a number of ad hoc routing protocols. The nodes auto-configure themselves as they move through a network composed of regions of nodes operating different routing protocols or they reconfigure themselves in response to changed networking conditions. As a result of the multi-state nature of the network-layer, heterogeneous, but contiguous, regions of nodes may be formed in which nodes operate incompatible routing protocols. Even though multiple regions of nodes may be within transmission range of each other, the heterogeneity of the ad hoc routing protocols partitions the network.

The following sections will detail the objectives and challenges that motivate this feature of the flexible networking system. The overall rationale underlying the design proposed in this thesis is then presented in Section 8.3. The specific operation of a scheme that addresses the problems associated with inter-regional networking is then presented, using the three candidate protocols to illustrate the difficulties that must be overcome, in Sections 8.4 and 8.5. The AODV, DSR and OLSR protocols are ideal candidates to illustrate and expose the issues that arise in solving this problem as each of their route discovery, route maintenance and packet-switching approaches are quite different, as detailed in Appendix A.

8.2 Ad Hoc Internetworking

8.2.1 Objective

The objective of an ad hoc internetworking component is to enable route discovery, route maintenance and, most importantly, packet-switching services to operate across a network composed of multiple heterogeneously configured regions of nodes. To preserve the robust characteristics of the constituent ad hoc routing protocols, the internetworking component must observe the specific distributed principles underlying the design of each of those constituent ad hoc routing protocols.

8.2.2 Challenges

Enabling internetworking between heterogeneous regions of nodes should not compromise the inherent characteristics of the underlying non-hierarchical ad hoc routing protocols. As with the auto-configuration and reconfiguration components, techniques to enable internetworking should also be designed in a manner that does not impact on the autonomy of the nodes or on the open and distributed nature of the network. A brief overview of existing approaches to the organisation and interconnection of heterogeneous subnets or networks illustrates that current techniques will not suffice.

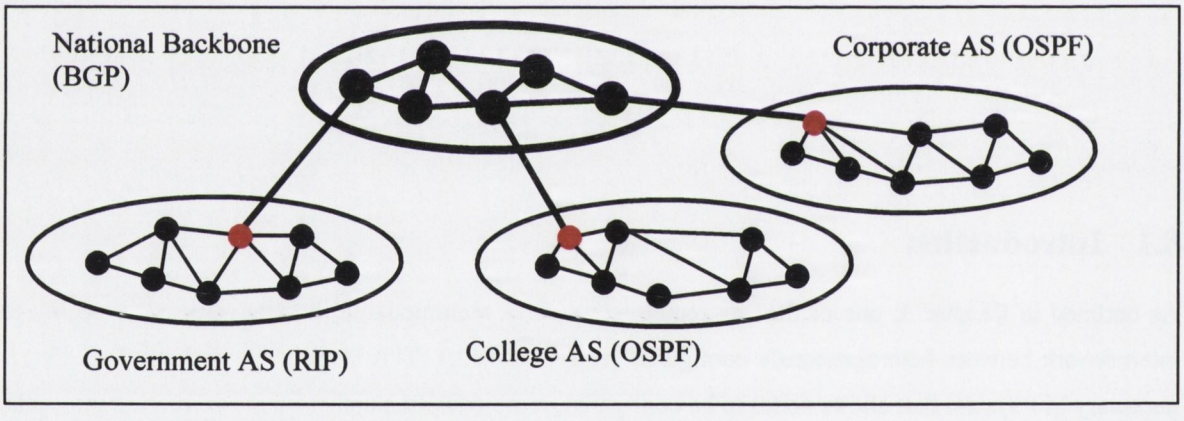


Figure 8-1 Hierarchical Internet Structure

Mobile wireless ad hoc network nodes differ from the routers and gateways found in the Internet's autonomous systems (ASs) in many respects. Typically, ASs consist of wired, static, manually configured routers and are connected to specific gateways that allow for communication between autonomous systems. The ASs are usually arranged in a hierarchical manner, using backbones to internetwork. Figure 8.1 depicts a typical hierarchical Internet system structure composed of three ASs using a variety of interior routing protocols. While one AS may use the Routing Information Protocol (RIP), another AS may use the Open Shortest Path First (OSPF) protocol. These heterogeneously configured domains are interconnected by means of a hierarchical backbone that uses a gateway protocol, such as the Border Gateway Protocol (BGP), to exchange routing information between one AS and another. BGP routing policies may be much more complex than those used by OSPF and RIP, more often than not including routing policies influenced by commercial or political factors. It should be noted that both the hierarchical backbone routers and the ASs use *fixed* and *dedicated* gateway machines, as depicted by the red nodes in Figure 8.1, to control

communication between the ASs and the backbone. These gateways provide a permanent point of contact between one AS and another, via the backbone.

Within the mobile ad hoc networking community there has been a suggestion that wireless gateways, using the Mobile IP standard [Per96],[Per96a], should be used to connect ad hoc networks to the Internet [Sun02], [Jon00], [Xu03]. The interconnection of mobile ad hoc networks and the fixed Internet has a different aim to that of the internetworking of standalone mobile ad hoc networks proposed here. The gateways described in [Sun02] allow ad hoc networks to internetwork to the Internet through designated Mobile IP Agents. Figure 8.2 depicts a static Internet gateway (the large red node) that has a wireless interface acting as a Mobile IP Foreign Agent. Ad hoc nodes that come within transmission range of the Foreign Agent may use Mobile IP techniques to access addresses within the Internet. This approach promotes the idea of a single homogeneous ad hoc network as an extension to the fixed Internet, whereas the internetworking proposed in this thesis remains within the confines of the mobile ad hoc networking context. It should be noted that both concepts are complementary; there is no reason why a flexible multi-protocol ad hoc network should not connect to the existing fixed Internet using the above-cited technologies.

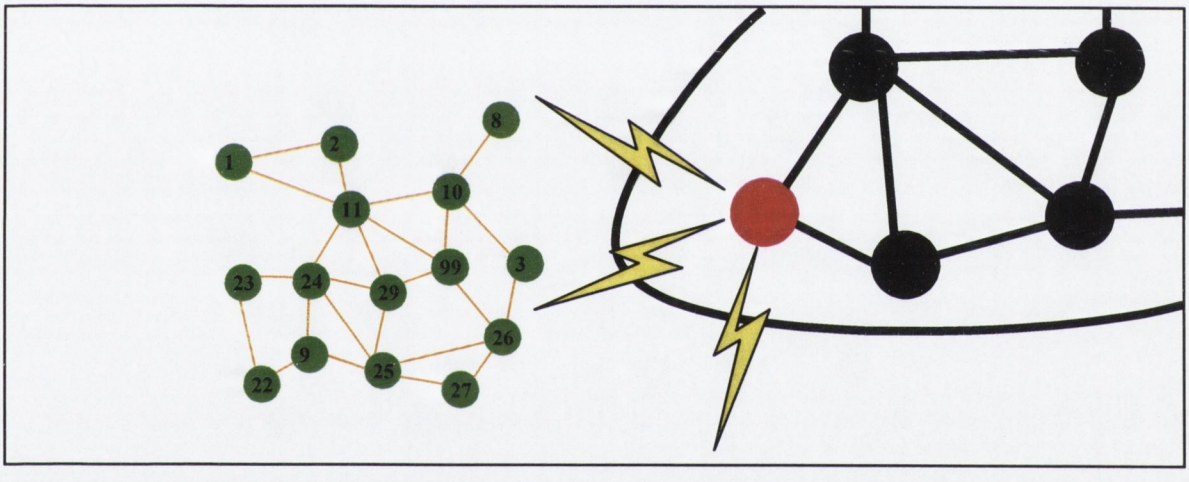


Figure 8-2 Internet to Ad Hoc Network Connections through Fixed Gateways

In an open mobile wireless ad hoc network the size and spatial scope of the network changes dynamically as nodes join or leave the network. A network may also partition as nodes move in opposing directions. In an ad hoc network that allows nodes to adapt to the perceived networking conditions in a distributed and regionalised manner, the network may decompose itself into a patchwork of contiguous regions of nodes operating different routing protocols. Consequently, as nodes move about in an autonomous manner, the *gateway* nodes that are on the boundary of the two heterogeneously configured regions of nodes may change. There is no permanence to the gateway nodes, as there is in the case of nodes that act as a gateway in the Internet structures as depicted in Figure 8.1 and 8.2; their position relative to other nodes and adjacent networks is inherently transient.

Within the context of the flexible multi-protocol network-layer proposed in this thesis, the autonomous mobility and heterogeneous configuration of nodes is facilitated by the auto-configuration and reconfiguration components such that the networking scenario illustrated in Figure 8.3 may arise. Figure 8.3 depicts a network composed of two regions of nodes at time τ_x. The region of green nodes on the left of the

diagram is using DSR and the region of blue nodes on the right is using OLSR. At time τ_x, Node 50 occupies a position that would allow it to provide a gateway between the two heterogeneous regions of nodes; enabling traffic to flow from DSR Node 25 to OLSR Node 66. However, given the mobile nature of the ad hoc network, a time later, at time τ_y, Node 50 has moved away from its former position as an intermediary and Node 7 now provides a similar point of contact between the two regions, enabling traffic to flow from DSR Node 25 to OLSR Node 66. So, not only are the DSR and OLSR regions of nodes independently mobile, but the point of contact between the two regions is also changing as result of that inherent node mobility.

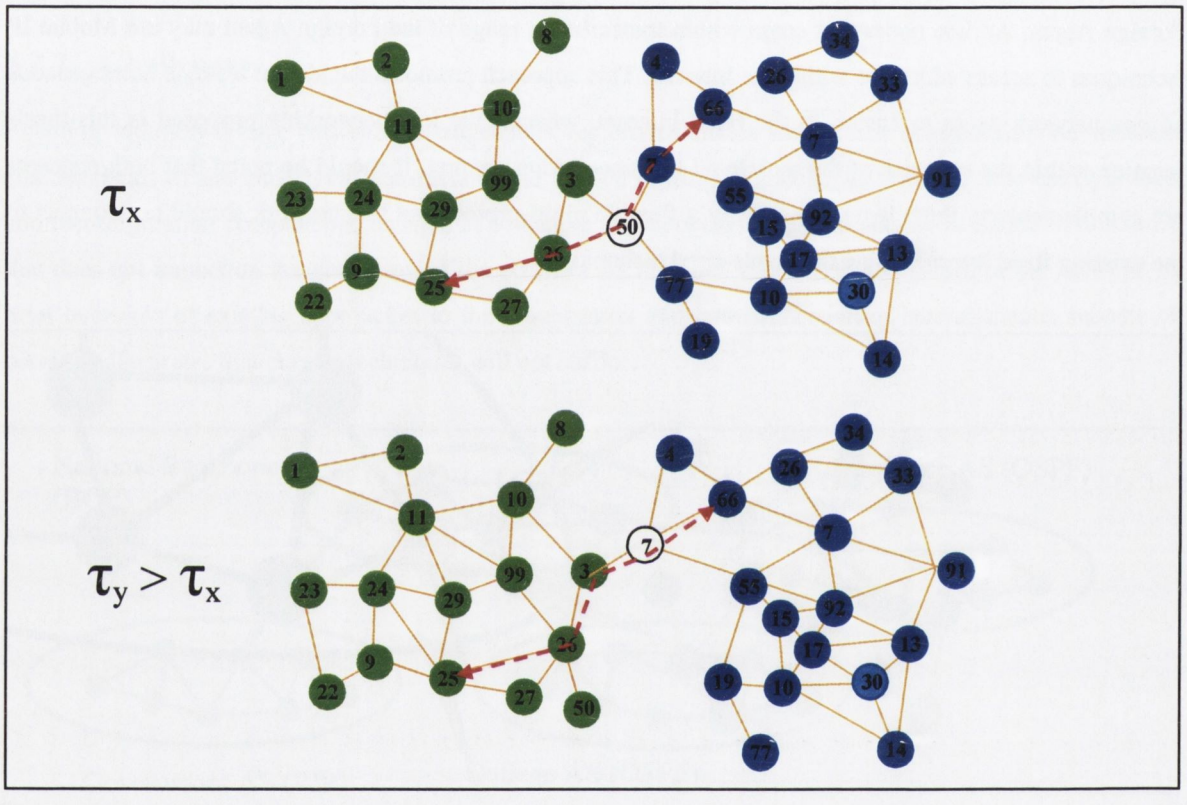


Figure 8-3 Multiple Heterogeneous Ad Hoc Networks Require Dynamic Interconnection

Therefore, to enable the objective of internetworking two mobile regions of heterogeneously configured nodes, the component must operate in the absence of fixed gateway nodes. The component should be able to use non-hierarchical techniques to the discover gateways that are temporarily set-up and torn-down as the configuration of the network changes over space and time.

8.3 Inter-Regional Multi-Protocol Routing

In order to design an internetworking component that respects the underlying characteristics of the routing protocols it seeks to meld together, it is insightful to consider the basic approaches taken by these protocols in carrying out their normal routing tasks within a homogeneous network, i.e. in a conventional single-protocol ad hoc network setting. The non-hierarchical routing protocols used in this thesis represent the distributed reactive and proactive classes of protocols. Both of these approaches cope with the dynamic and autonomous characteristics of ad hoc networks in different ways. In reactive protocols, such as AODV

and DSR, nodes discover routes when requested to do so by higher layers. They have no prior knowledge of the topology of the network. If the route changes after it has been established, then a new route is discovered or the broken one is maintained. This route maintenance generally also happens in a reactive manner. Conversely, proactive protocols seek to have a constantly updated view of the entire network topology regardless of any demand for routes. The effects of the autonomous mobility of the nodes, as reflected in the transient nature of the topology, are learned of in a proactive manner so that the route cache continually captures the state of the network's topology.

Extending the same philosophy to the task of heterogeneous routing, or internetworking, requires the incorporation of techniques that enable nodes to learn of, and store, information regarding routes that extend beyond the source nodes' own 'home' networks in an ad hoc manner that is in accordance with the nodes' normal routing policies. The techniques used in this component must maintain the important traits of each of the constituent ad hoc routing protocols; namely, they should ensure that the heterogeneous network is scalable, that it is free from routing loops and that its signalling profile corresponds to the signalling profiles of the underlying constituent routing protocols, i.e. the level signalling generated within any region should not be detrimentally affected by the heterogeneity of the entire network. Furthermore, the manner in which the networking conditions affect the routing protocols should not change, e.g. OLSR should not become inefficient at high densities when heretofore it was optimised for such conditions when operating as a standalone protocol.

In order to achieve these aims it is necessary to conceptually divide the network into Home and External Networks which are connected by means of Boundary Nodes, i.e. nodes which operate the protocols of the networks on either side of the region-to-region boundary. An internetworking component, the Broker, will control the transformation and the flow of control and data packets between the heterogeneous regions. Specifically, to ensure that the internetworking technique is scalable, the Broker must control the diffusion of broadcast packets across boundaries and it must control the infusion of routing information from one region into another heterogeneously configured region. Applying such techniques guarantees that scalability, freedom from routing loops and the existing signalling profiles are maintained. In order to elaborate on the approach that is taken in this thesis, some nomenclature is introduced at this stage.

8.3.1 Home Networks, External Networks, Boundary Nodes and External Routes

The concepts of Home Networks, External Networks, Boundary Nodes and External Routes are introduced so that it is possible to distinguish between the various elements of the global network system that feature in the internetworking of a multi-protocol ad hoc network. Figure 8.4a depicts a sample network of four contiguous but heterogeneously configured regions of nodes. The auto-configuration and reconfiguration components dictate the regional composition of the network, *vis-à-vis* its network-layer configuration. In particular, the auto-configuration component, as described in Chapter 6, detects when a node sits on the boundary of two regions and should concurrently operate the protocols of the two regions that it straddles. Such a node is a Boundary Node. For example, Node 3 in Figure 8.4a straddles Region I (AODV) and Region II (DSR) and concurrently operates both AODV and DSR.

From the point-of-view of Node 1 in Region I (AODV), the AODV nodes within this homogeneous region comprise its Home Network. Node 1 may use normal AODV techniques to find routes to any node in this region. Likewise, Node 6's Home Network is Region II (DSR) and Node 16's Home Network is Region III (OLSR).

Nodes distinguish between routes that begin and end within their own region, the Home Network, and routes that extend through a Boundary Node into another heterogeneously configured region of nodes, i.e. an External Network. A route that extends through a Boundary Node into an External Network is considered to be an External Route. From Node 1's point-of-view in Region I, a node that it discovers in any of the other regions, e.g. Region III, is regarded as an External Node. The configuration of that node does not concern Node 1. Source nodes are not required to record any information about the type of routes or regions that constitute the External Network, other than the cost, measured in hops, from the Boundary Node to the External Node; the fact that an External Route may traverse multiple regions within the External Network is of no consequence to the source node.

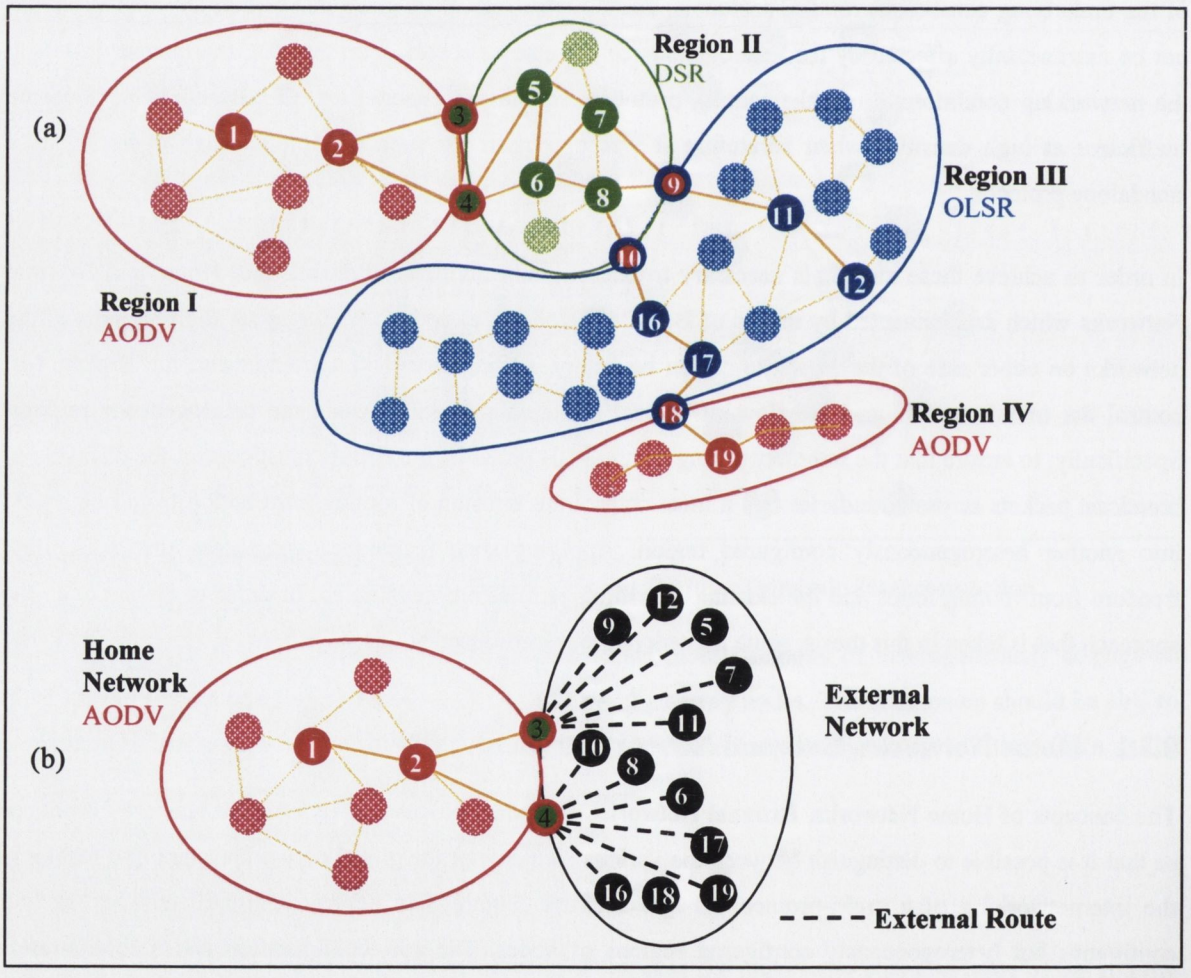


Figure 8-4 Home Networks, External Networks, Boundary Nodes and External Routes

To clarify, consider Node 1 which is communicating with Node 7 in another region. The route from Node 1 to Boundary Node 3 occurs within the source node's Home Network, i.e. Region I (AODV). However, the route extends from Boundary Node 3 to the destination node, Node 7, which is in another region or External Network. From Node 1's perspective, and indeed from the perspective of the intermediate nodes on the route

in that region, such as Node 2, the rest of the route, i.e. Node 3 to Node 7, is considered to be an External Route. Information about this route is kept in the External Route Cache of each node. More detail about the operation of the External Route Cache is presented later. The only information that Node 1 must record is the Boundary Node through which the External Node is accessible and the distance from the Boundary Node to the External Node, the External Distance. For example, consider the route from Node 1 to Node 16 in Region III, as depicted in Figure 8.4b. Node 1 simply records that Node 16 is an External Route accessible through Boundary Node 4 at a distance of 5 hops from the Boundary Node.

However, in order to gather such information in a manner that respects the underlying characteristics of the constituent ad hoc routing protocols it is necessary to leverage more information from the existing routing protocol control packets in addition to introducing some new structures, message fields and message processing techniques.

8.3.2 Brokering Inter-Regional Route Discovery, Packet-Switching and Route Maintenance: An Overview

As normal ad hoc network routing comprises the three phases of route discovery, route maintenance and packet forwarding, the internetworking component has to accommodate each of these phases if it is to enable heterogeneous regions of nodes to experience a seamless end-to-end unreliable packet-switching service across a network of the type exemplified in Figure 8.4a. While the route discovery, route maintenance and packet-switching phases of the three protocols are tackled by different mechanisms, they ultimately reduce to topology diffusion techniques and packet-switching techniques which, in their *intent*, have much in common. All routing protocol discovery and route maintenance phases, whether using a unified approach as in OLSR or a decoupled approach as in AODV and DSR, reduce to the issuing, receipt and processing of various kinds of *topology-information-request* messages and *topology-information-update* messages. `AODV_Route_Request` and `DSR_Route_Request` are examples of topology-information-request messages, i.e. messages that explicitly query routing information about a specific destination node. Topology-information-update messages include the `AODV_Route_Reply`, `AODV_Route_Error`, `DSR_Route_Reply`, `DSR_Route_Error`, `OLSR_Hello`, `OLSR_Topology_Control`, `OLSR_MIS` and `OLSR_HNA` messages, which contain routing information that may be used by all nodes regardless of whether the node has explicitly requested the routing information or not. A description of these messages and their use is given in Appendix A.

Within homogeneous networks, operating either AODV or DSR or OLSR, broadcast control packets are diffused from the source in waves according to protocol-specific rules. AODV and DSR nodes use expanding ring techniques and binary exponential back-offs to control the dissemination of broadcast packets whereas an OLSR node uses an optimised set of neighbours, the MPR set, to relay broadcast messages on its behalf. Each of the protocols handle the unicasting of packets, whether data packets or control packets, according to their own protocol-specific rules, in each case based solely on the packet's destination address and its TTL field. However, the diffusion of control packets comes to a halt when the packet reaches a node at the edge of the region as a control packet of one protocol is not intelligible to nodes operating other protocols. In short, an AODV control packet cannot be processed by a DSR node or an OLSR node, and vice versa.

Consequently, the diffusion of control packets cannot continue into another heterogeneously configured region without suitable modification. Referring back to Figure 8.4a, it is possible to see that if Node 1, which is in Region I (AODV), issues an `AODV_Route_Request` for Node 16, the packet will diffuse as far as Nodes 3 and 4. However, when these nodes rebroadcast the request, as mandated by AODV, Nodes 5 and 6 will receive the request but will not be able to process it. Effectively, the AODV control packets only have scope within the AODV region.

The technique proposed in this thesis enables a node to spread the *intent* of its control packet from one region into another in a process of cross-border propagation which is managed by the internetworking component residing in a Boundary Node; the component is referred as the Broker.

■ **Boundary Node Brokers**

This description serves to give an overview of the general function of the Boundary Node Broker. More detailed descriptions of the operation of the Broker are presented by way of detailed worked examples in Sections 8.4 and 8.5. By suitably leveraging the information contained in both the Network-Layer header and the information in the individual protocols' control packets, it is possible to extend the diffusion of broadcast control packets and the directed packet-switching of control and data packets from one region into another region. The actions and services of the Broker are transparent to all other nodes; packets that are issued by the Broker appear to other nodes as though they are packets that are simply being forwarded or repeated by a regular single-protocol node. The Broker does not initiate a route maintenance or route discovery process on behalf of another node, rather it enables an existing route discovery or route maintenance process to extend into a heterogeneously configured region. It is important to appreciate this distinction as it is a key element in the preservation of the existing signalling and scalability properties of the underlying protocols.

In describing the Broker component, the concepts of a Source Network and a Destination Network are introduced to describe the network regions that a Broker Node straddles; the distinction between Source and Destination Network depends on the direction in which the packet is travelling. The Broker must transform every packet that is received from the Source Network into an equivalent packet compatible with the protocol of the Destination Network, i.e. the network into which the modified packet will be sent. The Broker distinguishes between topology-information-request messages, topology-information-update messages and data packets. For example, Figure 8.5 depicts a simple two-region network. Node 1 is attempting to discover a route to Node 22 using the AODV protocol. When the `AODV_Route_Request` reaches Boundary Node 13, the Broker at Node 13 transforms the packet into the equivalent DSR topology-information-request message, i.e. a `DSR_Route_Request` message, before re-issuing the packet as though it was being repeated by Node 13 according to normal DSR procedures.

The transformation of data packets is relatively straightforward; the Broker simply modifies the Network-Layer header according to its stored routing information for the Destination Node and forwards the packet. As noted in Chapter 6, the Boundary Node operates two protocols and maintains a route cache for each of the regions to which it is connected, thus enabling it to switch packets between the two regions.

8.3.3 Controlling Diffusion into the External Network

In order to preserve the underlying scalability and signalling properties of each constituent routing protocol, the Broker must ensure that the manner in which packets are diffused into the Destination Network is compatible with the diffusion technique of that region. The preservation of the underlying scalability of each independent ad hoc routing protocol is key to ensuring that the distributed ad hoc internetworking technique is also scalable. Using a combination of an External Requests Table, the TTL (time-to-live) and Hop Count fields of the Network-Layer header and a common control-packet identifier, the Broker can control both the rate at which, and the extent to which, packets cross over into heterogeneous regions.

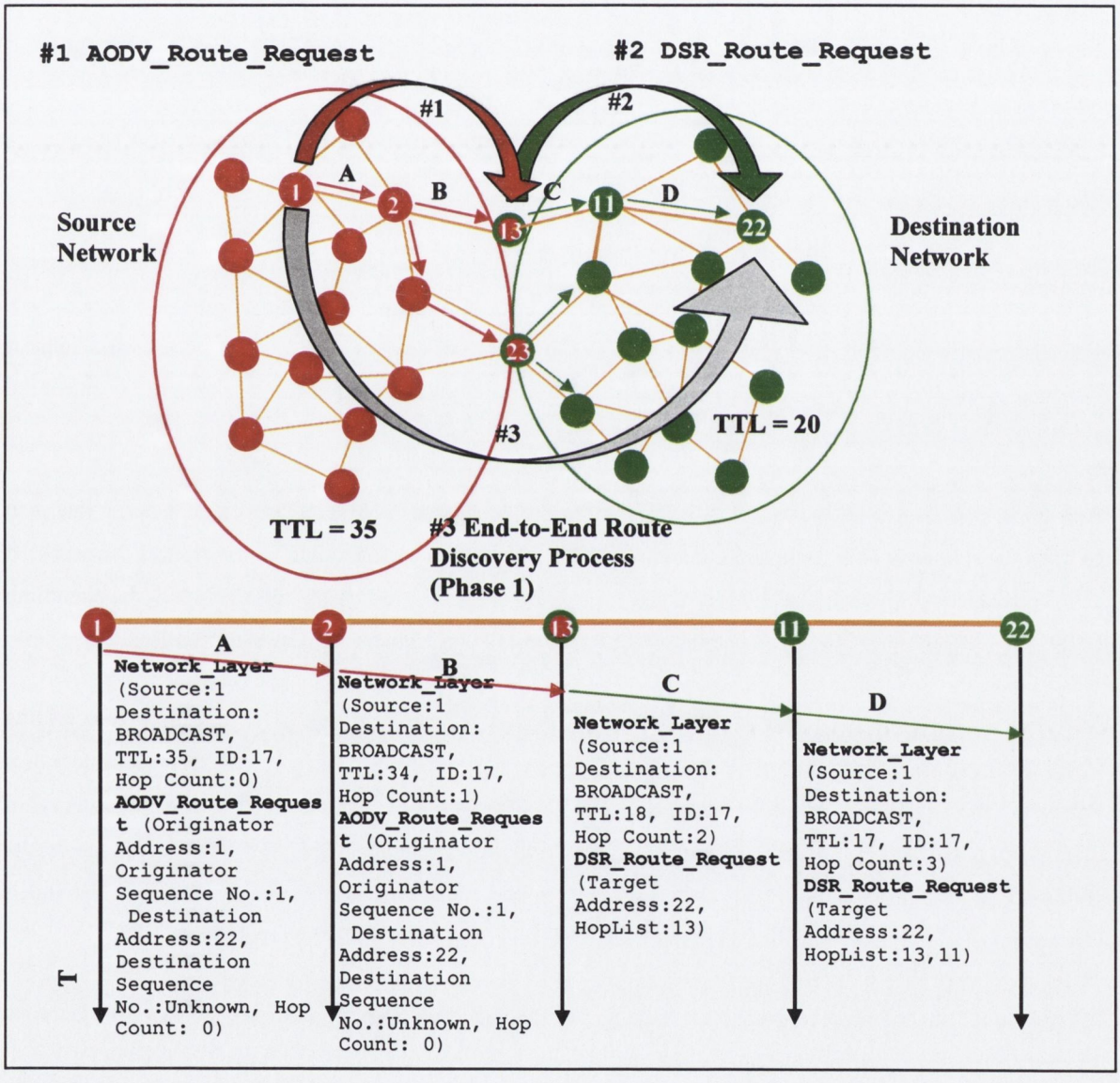


Figure 8-5 End-to-End Route Discovery Process: Phase One

■ **External Requests Table**

The External Requests Table has two functions. The first function is to limit the rate at which control messages are diffused across a region-to-region boundary. The second is to store a flag indicating that a node in the Home Network is awaiting information about a destination node; this function shall be described by means of the worked examples in Section 8.4.2. The External Requests Table is a modified version of the

FIFO Route Request Tables used in DSR networks. Every Boundary Node maintains a protocol-independent External Requests Table. The External Requests Table has seven fields; the Request Source field, the Destination field, the Consecutive Requests field, the Hop Count field, the TTL field, the Request Blocked Timer field and the Expunge Timer field, as depicted in Figure 8.6.

The Request Source field holds the address value of the source node that initiated the request process and the Destination field records the target address value of that request. The Hop Count and TTL fields hold the updated values contained in the Network-Layer header that encapsulated the request.

External Requests Table							
Entry	Request Source	Destination	Consecutive Requests	Hop Count	TTL	Request Blocked Timer	Expunge Timer
0	-	-	-	-	-	-	-

Figure 8-6 External Request Table

The rate-limiting purpose of the External Requests Table is analogous to the function of the Route Request Tables used in reactive protocols to control the rate at which broadcast request messages are issued by a node. In designing routing protocols with an aperiodic signalling content, i.e. where a node issues request messages as and when the demand arises, the protocol generally utilises a throttling mechanism to control the rate at which such messages are disseminated.

A problem arises when the Source Network's routing protocol issues broadcast packets at a faster rate than the Destination Network's routing protocol would issue them were it operating in the Source Network. If control packets are allowed to pass from region-to-region without being suitably throttled, then the signalling characteristics, and consequently the scalability, of the Destination Network will be compromised.

For example, consider the network depicted in Figure 8.5. If nodes in the AODV region were issuing `AODV_Route_Request` packets using a binary exponential back-off technique with an initial interval of 500ms, but the DSR network limits the rate of topology-information-request messages to a longer interval of 1000ms, then the Broker must throttle the rate at which it allows such messages to diffuse across the border from one region into another. Otherwise, the signalling characteristics of the Destination Network's network-layer, which are controlled by parameters of its ad hoc routing protocol, would be altered.

When a Boundary node receives a request message, which will have been broadcasted, it checks the External Requests Table to ascertain whether or not it may forward the request or whether it must block the packet. If no entry for the source/destination pair exists in the table already, the Broker transforms the packet in the normal way and then creates an entry for the request. In creating the entry, the node records the Source Address in the Request Source field and the targeted address in the Destination field. It also records the Hop Count and the TTL. The Broker sets the Consecutive Requests flag to 1. The Request Blocked Timer is set using the Destination Network's routing protocol's own request throttling mechanism. For example, Node 13 in Figure 8.5 would use the DSR throttling technique and would set the timer to 1000ms in accordance with

the DSR settings for a first request attempt. The Expunge Timer is set to at least the maximum round-trip time that a request may take to return to the node, based on the estimated hop-to-hop time and the modified TTL value, i.e. 2 x Node Traversal Time x modified TTL. The TTL modification is explained in the next section.

When another request with the same source/destination pair is received, the Broker performs the following steps. If the Request Blocked Timer has not expired, the Broker Node delays the request at the node until the timer expires. The Consecutive Request field is incremented and the Expunge Timer is set again. If the Destination Network's routing protocol is using a binary exponential back-off, then the value of the Request Blocked Timer may be doubled as it is the second attempt. If the Blocked Timer has expired when a second request is received, then the broker just resets all of the entries in the table and lets the request through. If another request is received for the same source/destination pair while the Broker is already holding back another request, then the latest request is silently dropped. This rate-throttling mechanism ensures that one region cannot flood another region with more control messages than its own protocol allows its own nodes to issue themselves.

■ TTL and Hop Count

The second element of the process that the Broker must invoke when controlling the diffusion of control packets ensures that the scope of a packet, as dictated by the TTL field of the Network-Layer header, is in accordance with the maximum TTL value specified by the protocol in the Destination Network. The TTL value limits the lifetime of a packet. In conventional networks the TTL field is used to quench packets which may get into routing loops or other arbitrary cycles so that they do not waste excessive network resources. While this application of the TTL also holds true in mobile ad hoc networks, the TTL also serves another purpose, that of limiting the extent of the route discovery or topology update processes.

As the topology of an ad hoc network is subject to change, ad hoc routing protocols limit the length of routes that may be discovered by mandating that the Source Node sets the TTL value to an 'appropriate' value, thus limiting the effective diameter of the network when viewed as centred about the source node. If the TTL is too large, nodes may engage in route discovery cycles which are too ambitious, and consequently costly, for the type of underlying networking conditions. If a node seeks to establish a long route, say 20 hops, in a very mobile network, such a request may never be satisfied due to the incompatibility of the route discovery technique being used and the networking conditions being experienced. The TTL value may be set on a network-wide basis or on a node-by-node basis, depending on the ability of the underlying ad hoc routing protocol to adapt its local parameters to the prevailing network conditions. Regardless of whether the TTL value is statically or adaptively set on a node-by-node or network-wide basis, the Boundary Node uses the maximum allowed TTL value of the Destination Network's routing protocol to limit the scope of the packets being diffused into it from the Source Network.

To explain, consider two adjacent regions of nodes which are heterogeneously configured, such as the network depicted in Figure 8.5. The AODV region sets a network-wide maximum initial TTL value of 35 hops while the DSR region sets a network-wide TTL value of 20 hops. When Node 1 issues an `AODV_Route_Request` message, it is flooded throughout the AODV region with an initial TTL value of

35. This field is ordinarily decremented at each intermediate node. When the request message reaches the Boundary Node 13, the remaining TTL value is 34. If the Broker were to simply transform the intent of this message by issuing a `DSR_Route_Request` message with TTL 33, then the DSR request issued by this node would travel further than a request initiated within the DSR region itself. Such an unmodified message would compromise the characteristics of the DSR region as such a request message may discover a route which is too long to be efficiently supported by the DSR protocol within that region.

In order to curtail the extent of the cross-border flooding of this packet, the Broker makes an extraordinary modification of the Network-Layer header by reducing the TTL to 20 from 34, 20 being the TTL value that the Boundary Node would use were it originating a request itself. However, it then has to take account of the number of hops that the request has already expired in reaching the Boundary Node, as indicated by the Hop Count field. In this case the request has a Hop Count of 2 at Node 13 so the new TTL of 20 is further reduced by 2 to 18. This modification has the effect of creating a packet that appears to have been originated with an initial TTL of 20, as would be true for a packet originated by DSR Node 13. In performing this modification, the Broker ensures that the message conforms to the Destination Network's routing protocol parameters. The modification is shown in the Network-Layer headers of messages B and C, illustrated in Figure 8.5.

In sum, the effects of this modification of a broadcast-diffusion packet are two-fold. Firstly, the technique ensures that the internetworking technique preserves the scalability of the Destination Network's ad hoc routing protocol. Secondly, as a consequence of preserving that scalability the technique ensures that the source node in the Home Network only learns of routes that the External Network can support according to its own interpretation of the networking conditions.

■ **Common Control Packet Identifier**

The final element of the process of controlling the diffusion of control packets across region-to-region boundaries involves the modification of the unique identifiers that the protocols use to control the quenching of broadcast packets.

In conventional networking systems, each of the three protocols uses a unique packet identifier to stem the dissemination of their broadcast packets. For DSR, the `DSR_Route_Request` message is the only message that is broadcast throughout a DSR network. The `DSR_Route_Request` message contains a 16-bit unique identifier. The flooding of `DSR_Route_Request` messages is quenched based on the Network-Layer header's Source Address of the message and the unique identifier in a similar manner to the AODV approach. Details of these control messages are described in Appendix A.

Since OLSR is a proactive link-state protocol in which a few different messages are flooded regularly throughout an OLSR network to enable the construction of complete routing tables, the protocol places a 16-bit unique identifier in each of the messages that are broadcast. `OLSR_TC_Messages`, `OLSR_MIS_Messages` and `OLSR_HNA_Messages` are flooded throughout the network. OLSR quenches the flooding of such messages based on the Network-Layer header's Source Address and the unique identifier number of each message.

`AODV_Route_Request` messages are flooded throughout an AODV network. Each `AODV_Route_Request` contains a unique 32-bit identifier, the RREQ_ID. As `AODV_Route_Request` messages are flooded through the network, intermediate nodes quench the flooding of the messages based on the Originator Address specified in the control message and the RREQ_ID. It should be noted that the other protocols used the Network-Layer Source Address.

As the messages are translated from one protocol to another by the Broker component, it is desirable that the unique identifier of the broadcast message is maintained into the new region. The need for this feature arises where a broadcast message enters another region through multiple Boundary nodes. Each Boundary Node should create a new message which is identifiable as having emanated form the same source node. To resolve this issue, all packets are quenched on the basis of their Network-Layer Source Address and Packet Identifier fields which are uniquely set by the source node regardless of the routing protocol in use. Referring to Figure 8.5, it may be seen that the `AODV_Route_Request` message enters the DSR region through two Boundary Nodes. However, even though the request is now in the form of a `DSR_Route_Request` message, the nodes in the DSR region can still quench the second request as it uses the same unique identifiers.

8.3.4 Infusing External Routing Information into the Home Network

Another key purpose of the Broker component is to control the infusion of routing information relating to an External Route, which extends into an External Network, back to nodes in the Home Network. In order to infuse such information into the Home Network, the Broker abstracts the routing information pertaining to the discovered External Route. Ordinary nodes in the Home Network only record abstracted information about External Networks in their External Route Caches. The External Route Cache is described next, followed by a description of the technique used to infuse external routing information into the Home Network. The issue of routing loop avoidance in the presence of such heterogeneous routing information is also discussed.

■ External Route Cache

Every node maintains a protocol-independent External Route Cache in addition to its own normal, protocol-prescribed route cache. The distance-vector based External Route Cache has three fields; the External Node field, the Boundary Node field and the External Distance field. The structure is depicted in Figure 8.7. In order to facilitate ad hoc internetworking, the distance-vector concept is modified to have the following meaning. The destination referred to by the distance-vector refers to the External Node, i.e. the destination node that lies in an External Network and which was originally sought by the source node. The next-hop field refers to the Boundary Node, a node within the source node's region which offers access to the External Network. The distance refers to the distance, in hops, from the Boundary Node to the External Node.

To clarify, consider again the sample network illustrated back in Figure 8.4a. The source node, Node 1, has a route to Node 7. The route from the source node, Node 1 to the Boundary Node, Node 3, falls within the AODV region of the network. This route appears as a normal DSR route cache entry, as depicted in Figure

8.4a. The route from Boundary Node 3 to the destination in the DSR region, Node 7, is an External Route. This route appears in the External Route Cache.

AODV Distance-Vector Cache (Node 1)				
Entry	Destination	Next Hop	Distance	Type
0	3	2	2	Forward

AODV External Route Cache (Node 1)			
Entry	External Node	Boundary Node	External Distance
0	7	3	2

Figure 8-7 External Route Cache Structure

If Node 1 required a route to Node 7, it would query its DSR route cache first. Since this request would fail, Node 1 would then query its External Route Cache which would indicate that Node 7 is accessible through Boundary Node 3. As Node 1 has a route to Boundary Node 3, it may route data to Node 7. The entries recorded in the External Route Cache are managed in conjunction with the corresponding entries in the node's regular route cache. When normal routing protocol-specific mechanisms expunge entries from a node's route cache, dependent External Routes are also removed. Explicit route maintenance techniques, described in Sections 8.5, enable route maintenance information to be relayed from one region to another when appropriate. Sections 8.4 and 8.5 provide more specific detail about the use of the External Route Cache.

■ External Route Infusion

As noted, the Broker Node is responsible for controlling the infusion of external routing information back into the Home Network. Referring to the example network depicted in Figure 8.8, it may be seen that the second stage of a route discovery process, which extends across an AODV-DSR network, involves the relay of a `DSR_Route_Reply` message back to the source node, Node 1. When the reply reaches Boundary Node 13 the scope of the DSR protocol ends. The source-routing information contained in the `DSR_Route_Reply` message cannot be injected into the Home Network as is. Rather, it is necessary to abstract the pertinent information from the `DSR_Route_Reply` message before infusing it into the Home Network. Distance-vector routing information is the most abstracted form of routing information used in contemporary networks. Unlike link-state routing or source routing, distance-vector routing information reveals very little information about the actual route from source to destination. As noted above, the distance-vector simply indicates the cost of reaching the destination node, normally in hops in an ad hoc network, and it identifies the next-hop node that should be used to reach that destination.

This External Route information, in the form of a distance-vector, must be conveyed to the source node using topology-update messages that are understood by the protocol of the Home Network. Both DSR and OLSR provide for the identification of External Networks and External Routes using topology-update messages. In the case of OLSR External Networks are identified using the `OLSR_HNA` message, described in Appendix A. DSR provides for the identification of External Networks through the use of flags in its `DSR_Route_Reply` and `DSR_Source_Route` messages. The DSR specification does not specify the interaction of the protocol with External Networks but it does, as with OLSR, recognise that the

internetworking of a DSR network and another network is a possibility. Specifically, the External Network flags in the DSR messages allow for the identification of nodes that lead to External Networks, i.e. Boundary Nodes. The interaction of DSR with 'gateway' nodes and External Networks is not specified or expounded upon in the literature. The `DSR_Route_Reply` message is modified to include an extra Distance field if the Last Hop External flag has been set. This field accommodates the inclusion of External Distance information.

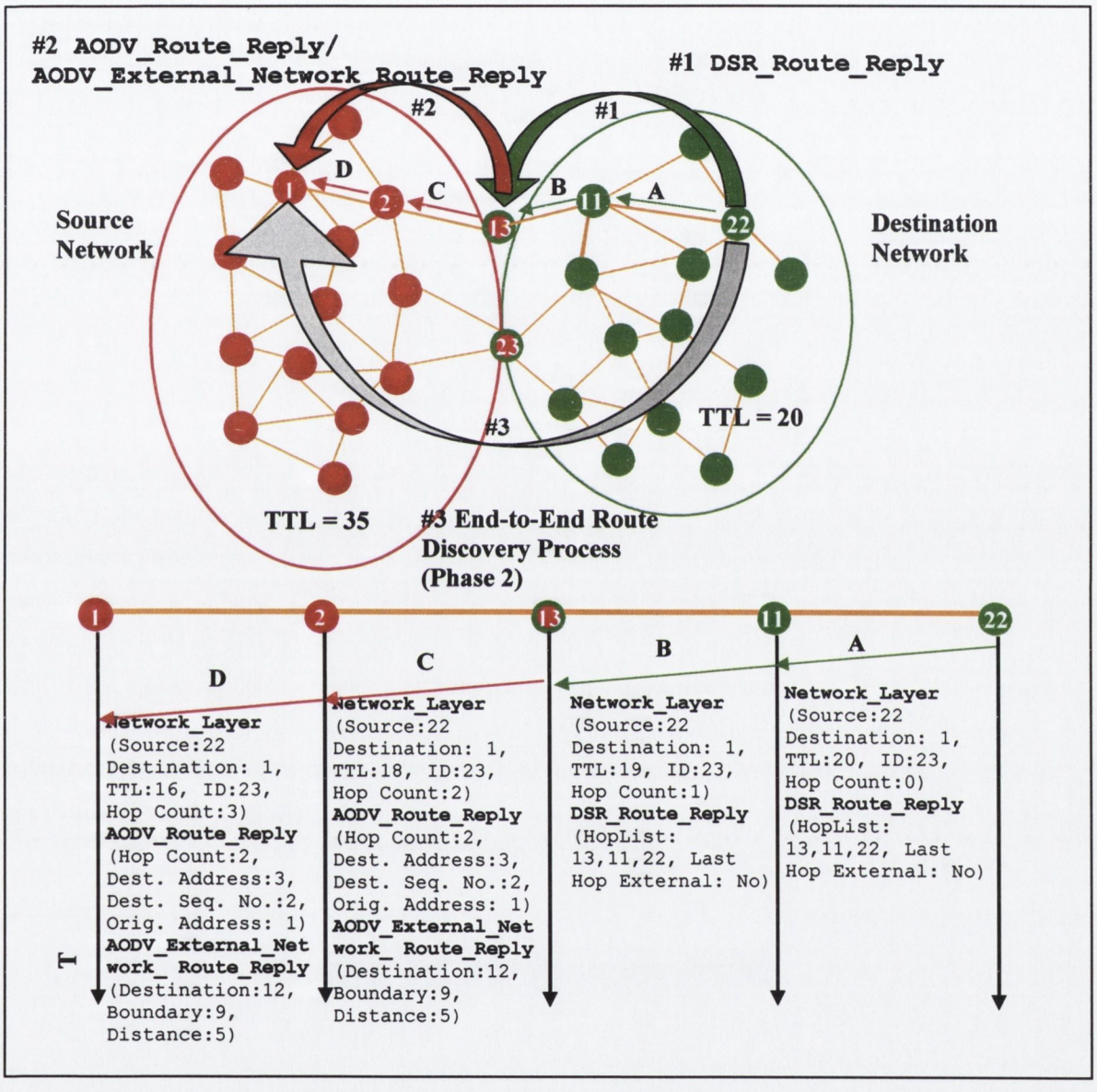


Figure 8-8 End-to-End Route Discovery Process: Phase Two

However, in the case of AODV there is no provision made for the identification of Boundary Nodes or conventional 'gateway' nodes in the manner that both DSR and OLSR provide. The AODV specification discusses the interaction of subnets in which each subnet has a routing prefix but does not provide any specific recommendations or suggestions. This approach is based on the assumption of hierarchical addressing structures that are prevalent in conventional networks. The AODV protocol does however allow for the use of unspecified extensions to the `AODV_Route_Request` and `AODV_Route_Reply`

messages. In order to describe the existence of External Networks and External Nodes, it is proposed that an **AODV_Route_Reply_External_Network** extension message be defined.

The format of the **AODV_Route_Reply_External_Network** extension message is illustrated in Figure 8.9. The message contains the following fields; an External Node field, a Boundary Node field and an External Distance field. Again, Sections 8.4 and 8.5 provide more specific detail about the use of these messages and message extensions within the context of the operation of the internetworking component.

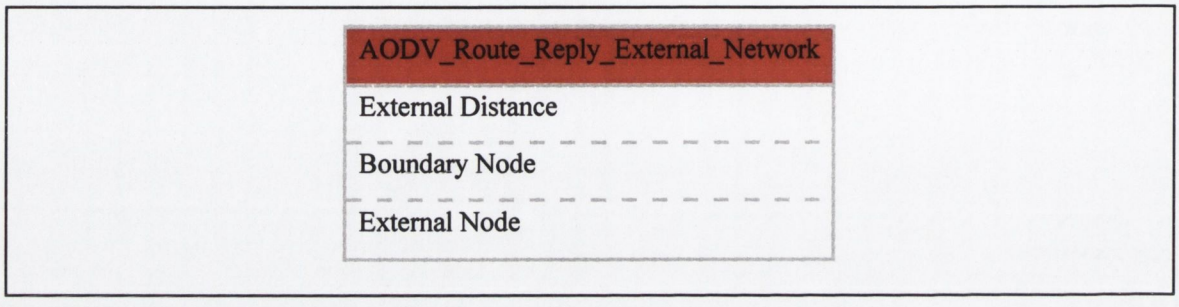


Figure 8-9 AODV_Route_Reply_External_Network Message Format

■ **OLSR: Associated Hosts and Networks**

OLSR is a proactive protocol and, as such, any node in the network is aware of every other connected node in the OLSR network at any given time. The OLSR specification also provides an extension that allows for the identification of associated hosts and networks. The OLSR specification acknowledges that certain nodes may act as gateways to external networks that do not operate the OLSR protocol. As with the other candidate protocols, the authors of OLSR designed the network based on the assumption of a hierarchical internetwork structure which would involve the use of hierarchical addressing structures. Taking such a view, they proposed that a node which has associated hosts or networks should periodically issue Host and Network (**OLSR_HNA**) messages. These messages are issued by each gateway node that offers connectivity to another non-OLSR network and list the network/host addresses and the network prefixes. Each node within the OLSR network is then able to incorporate this information into its route cache.

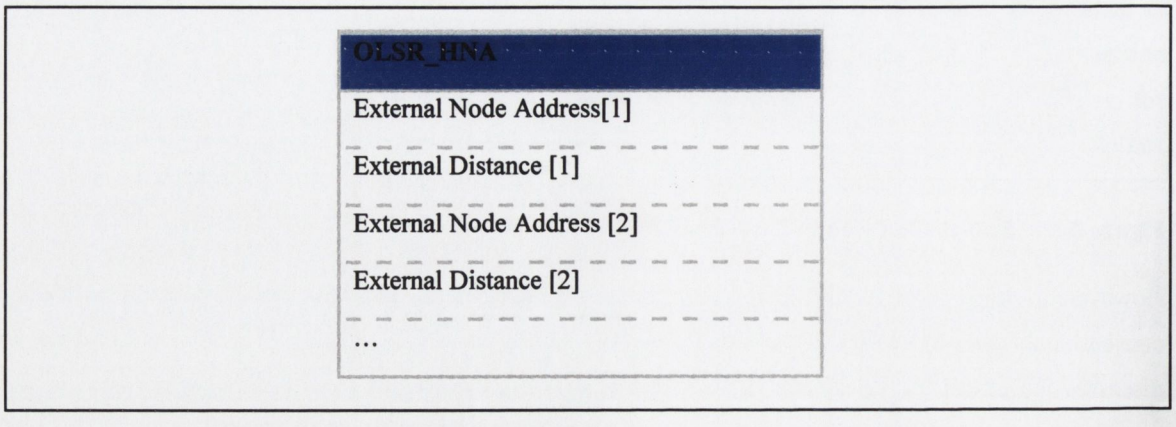


Figure 8-10 Modified OLSR HNA Message Format

In this system, a Boundary node is equivalent to a conventional gateway node. In this system, each Boundary Node operating the OLSR protocol issues **OLSR_HNA** messages that describe the known nodes (hosts) in the

other network to which the Boundary Node is connected, subject to the limits of the OLSR protocol. However, since protocol-specified **OLSR_HNA** messages deal with hierarchical addressing, the message must be slightly modified for use in this system, in detail rather than intent. Specifically, the network/host address field now records the External Node Address and the network prefix field records the External Distance to the External Node from the Boundary Node, as illustrated in Figure 8.10.

The combination of the regular OLSR topology updates and the **OLSR_HNA** messages means that any connected OLSR node is aware of the existence of every node in the OLSR network and it is also aware of the existence of Boundary Nodes that offer connectivity to External Networks that are not operating the OLSR protocol. Boundary Nodes are identified as those nodes that originate the **OLSR_HNA** messages.

When a data packet is presented to the networking layer in a node operating the OLSR protocol, the data forwarding mechanism would normally inspect the route cache for a valid route to the specified target address. If a route exists, the data packet would be forwarded to the next hop in the route to the target address. In the absence of a valid entry in the OLSR-maintained route cache, the data packet would be dropped. However, circumstances may arise in which the node bearing the target address requested by the higher layer is situated in the External Network, unbeknownst to the OLSR nodes. Each OLSR node can determine whether or not it is connected to an External Network if it has a single entry in its External Route Cache; this cache lists each of the Boundary Nodes to which the OLSR network is presently connected.

The flexible system necessitates a modification to the OLSR protocol to enable it to fully internetwork in a multi-protocol environment. This system uses an **OLSR_HNAQ** message to enable the OLSR network to trigger a search of its External Networks, if any exist. The proactive nature of the OLSR protocol coupled with the fact that each node has total knowledge of the connected network topology results in every other node learning of a successful response to the issuing of an **OLSR_HNAQ** message. An OLSR node's awareness of its region is only limited by the regular network constraints such as the allowed network diameter (TTL) and the connectivity of the nodes. If an OLSR node's request for an External Route is successful, then the entire network will learn of the route that leads to the External Node.

An OLSR node may attempt to initiate a search of the External Network if it receives a packet from a higher layer for which it does not have a valid route or if, as a Boundary Node, it receives a request from another region. The OLSR node does not buffer the data packet pending the discovery of External Route information into the OLSR network. If the OLSR network is experiencing sporadic traffic loading, then a subsequent route demand from the initiating node or any other node may be satisfied by its query of the External Network.

The format of the **OLSR_HNAQ** message, the OLSR Host and Network Association Query message, is identical in form to the **OLSR_HNA** message except that the destination specified in the External Node Address field is unknown to the network, i.e. it is the target address. If the **OLSR_HNAQ** message is originated by an ordinary OLSR node, then the External Distance field is set to zero. If the message is issued by a Boundary Node as a result of a query from another region, then the External Distance field is used to

reflect the distance between the original source of the request and the Boundary Node. Each time that a node forwards the message the Distance value is incremented.

OLSR_HNAQ messages are sent in the empty portions allotted for **OLSR_HNA** messages. When a node wishes to originate an **OLSR_HNAQ** message it queues it for the next periodic **OLSR_HNA** broadcast. If the OLSR node issuing the request is a Boundary Node that is using its full **OLSR_HNA** message space, then the **OLSR_HNAQ** message is not sent. **OLSR_HNA** messages may be issued on a periodic basis in accordance with the OLSR-specific rules. The use of **OLSR_HNAQ** messages does not alter the signalling characteristics of OLSR. By limiting **OLSR_HNAQ** messages to any space left in **OLSR_HNA** messages, this technique ensures that the use **OLSR_HNAQ** messages cannot add to the OLSR overhead anymore than **OLSR_HNA** messages already do.

Furthermore, each OLSR node maintains an External Request Table, as described in Section 8.3.3, which is used to limit the rate at which **OLSR_HNAQ** messages are issued by the network. Since the response to a query of the external network will be infused into the whole network, and not returned to the specific node which initiated the query as is the case of the reactive protocols, the OLSR network treats each **OLSR_HNAQ** message as if it has been issued by the network and not just the node that did issue it.

Each OLSR node uses the typical binary exponential back-off mechanism to control the rate at which the network issues the query messages. Before a node attempts to originate or rebroadcast an **OLSR_HNAQ** message, it checks the External Requests Table for a Destination entry bearing the same value as the External Node Address field specified in the message. Unlike AODV and DSR, OLSR nodes to not record the source address of the request as it is the network that is issuing the request through its Boundary Nodes. Also, if a forwarding node has no knowledge of a Boundary Node it drops the **OLSR_HNAQ** message as it will not reach an External Network.

The effects of this technique are that the network only ever engages in one search for each destination, rather than engaging one search for each querying node. This allows more **OLSR_HNAQ** messages to be carried in the **OLSR_HNA** message space.

Also, if a node receives an **OLSR_HNAQ** message specifying a destination for which it has a route, the receiving node drops the message without propagating it any further. The reason for a discrepancy between nodes regarding their topology knowledge may be accounted for by the stability of their connectivity to other nodes and their distance from the specified destination. If the querying node is beyond the allowed network diameter, then it will never receive routing information regarding the destination nodes regardless of any effort to stimulate an external query.

Once an **OLSR_HNAQ** message arrives at a Boundary Node it is treated as a topology-request-message in accordance with the provisions that have been described already in Section 8.3.3. and which will be further clarified by example in Section 8.4.2.

■ **Avoiding Routing Loops**

As the constituent ad hoc routing protocols of this system have been designed to guarantee freedom from routing loops within a homogeneously configured standalone ad hoc network, it is necessary that the ad hoc internetworking technique also ensures that routing loops are not formed in a heterogeneously configured network. A routing loop occurs if stale and incorrect routing information is allowed to persist in the network's routing tables such that nodes cannot find a route to a destination node as the routing information directs them to incorrect next-hop nodes.

The scenario which may be anticipated to occur involves a network composed of two regions; Region A operating Protocol A, Region B operating Protocol B. Nodes in Region A may be aware of routes to nodes in Region B which are recorded as External Routes. At some stage an External Node in Region B migrates to Region A. The nodes in Region A may discover this node while they still have information recorded about the External Node. Nodes always treat routing information collated by their own routing protocol as being more up-to-date than routing information acquired from an External Network.

To avoid routing loops, when a node learns of a route to a destination address within its own Home Network, by means of its own protocol, then the node removes any entries for that node in its External Route Cache. Furthermore, whenever a node receives information about an External Node in the form of an External Route, it only creates an entry or updates an old one if the Home Network portion of the routing information in the topology-update message is accepted as being valid by the Home Network's routing protocol.

8.4 Inter-Regional Route Discovery

As the ad hoc internetworking process is a complex one, which may involve a number of different scenarios, presentation of the techniques is most clearly done by means of worked examples. Two scenarios are presented in this section covering every region-to-region combination that may occur during the route discovery and topology-update diffusion phases of the internetworking process.

Route discovery is the first phase of a dynamic routing protocol to be invoked, regardless of whether it is reactive or proactive in nature. Within proactively operated networks, route discovery occurs immediately and continuously through the use of the neighbourhood sensing and topology-update diffusion techniques. In reactive protocols, targeted route discovery mechanisms are invoked when a node has a demand for a specific route. Proactive protocols provide total information, insofar as the underlying characteristics of the network allow, whereas reactive protocols provide partial, but specific, route information, subject to the same network constraints.

When data packets are generated by higher level layers in the communication stack they may have unicast, multicast or broadcast addresses. In the case of broadcast addresses, the packet is flooded through the network subject to packet quenching mechanisms at the network-layer. The route discovery mechanism and route cache of the ad hoc routing module of the network-layer are not invoked for broadcast addresses. The use of multicast addresses is outside the scope of this thesis as the candidate protocols only provide for the

route discovery and route maintenance of unicast-based routes. Other ad hoc routing protocols and protocol extensions are being developed and standardised to support multicast routing.

If the packet is a unicast one, then the services of the ad hoc routing protocol are called upon. The unicast packet will specify a single destination node by passing the required Destination (or Target) Address to the network-layer along with the higher layer's data packet. No matter which ad hoc routing protocol is in use, the Route Cache will be queried for a valid route to the specified Destination Address.

8.4.1 Route Discovery: Scenario One

In the first scenario, as depicted at the top of Figure 8.12, a network is composed of three regions of nodes. An AODV region is adjacent to a DSR region which is adjacent to an OLSR region. Figure 8.12 also provides an illustration of the messages that are passed between the nodes in this example, corresponding to the description provided below. The Network-Layer headers only depict fields used in this process; the node-to-node fields are not shown. Node 1 tries to discover a route to Node 12 for a data packet that has been passed to the network-layer from a higher layer. The AODV protocol places a copy of the data packet in its FIFO Send Buffer and then begins the route discovery process. None of the reactive nodes in this example have cached routing information at this time.

In this scenario, the AODV region is the Home Network and the DSR and OLSR regions form the External Network. However, from the DSR region's perspective, when it diffuses the request control packets into the OLSR region, it too is a Home Network and the OLSR region is an External Network. Consequently, the techniques described in the last section regarding the diffusion of control messages and the infusion of routing information will be explored at each of the region-to-region boundaries.

As an aside, it should be noted that if a Boundary Node receives data from a higher layer for which it does not currently have a valid route it would separately invoke the route discovery mechanisms of each of its protocols. A Boundary Node operating both AODV and DSR only operates one FIFO Send Buffer, i.e. it holds one copy of the data packet pending the discovery of a route through either region.

1.1 Node 1 originates a route request, **AODV_Route_Request(**`Originator Address: 1, Originator Sequence No.: 1, Destination Address 12, Destination Sequence No.: Unknown, Hop Count: 0`**)** subject to its own controls as set out in Appendix A, encapsulated in a Network Layer Packet with the following header values; **Network_Layer(**`Source Address: 1, Destination Address: BROADCAST,TTL: 20, ID: 17, HopCount:0`**)**. The Packet ID is set by the node to uniquely identify the packet, regardless of its contents. The Sequence Number of the routing protocol packet header is set by the AODV protocol's regular methods. The TTL specified in the **Network_Layer** header indicates the maximum network span that the AODV network allows; in this region it is set to 20. The Route Cache and External Route Cache for Node 1 are empty at this stage, as shown below.

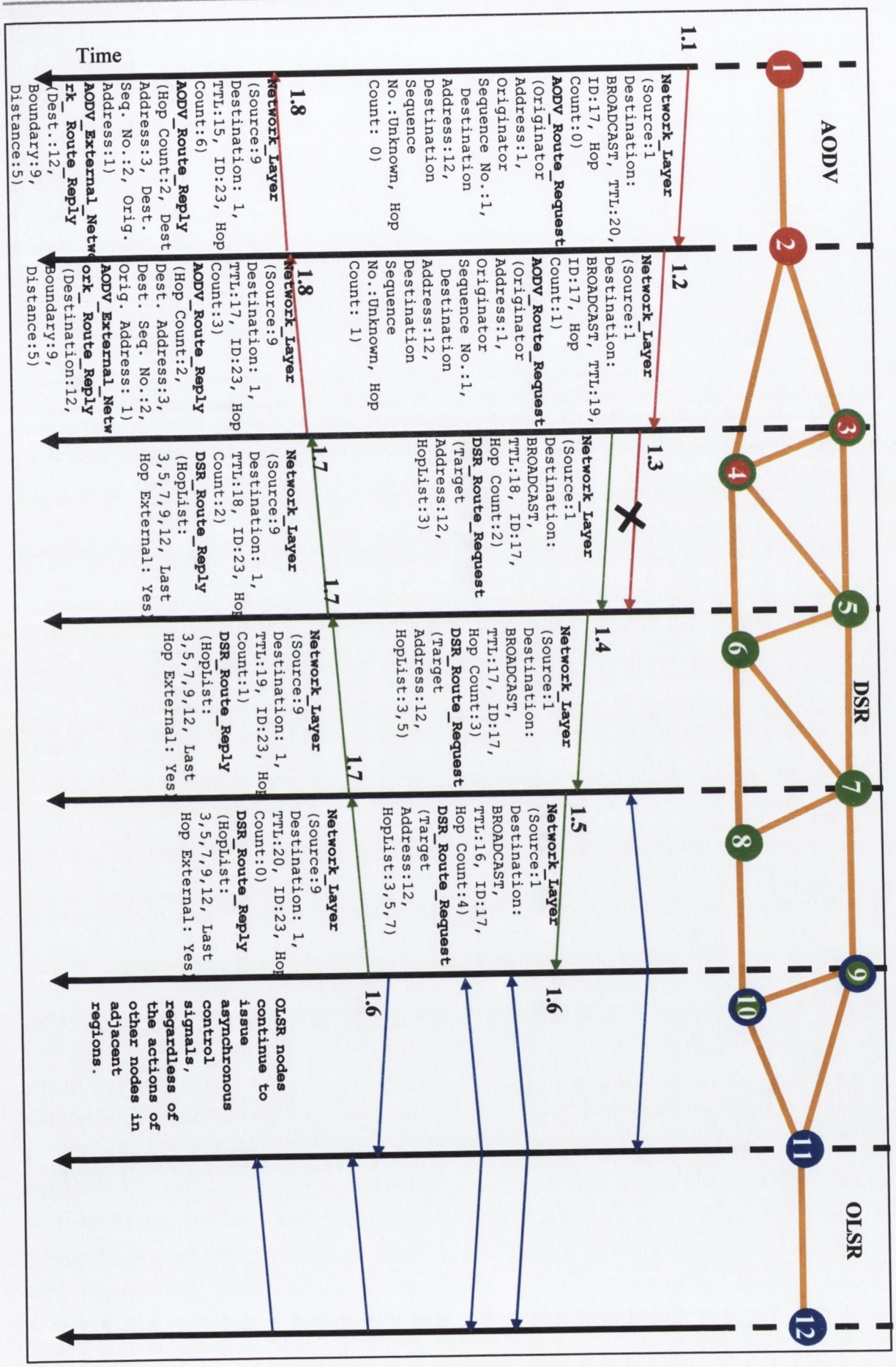


Figure 8-12 Scenario One: Network Diagram and Route Discovery Messages

AODV Distance-Vector Cache (Node 1)				
Entry	Destination	Next Hop	Distance	Type
0	-	-	-	-

AODV External Route Cache (Node 1)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

1.2 Node 2 receives the **AODV_Route_Request**, processes it, and rebroadcasts it since it cannot originate an **AODV_Route_Reply** itself. The Hop Count in both the request and the **Network_Layer** is incremented by 1 and the TTL of the **Network_Layer** header is decremented by 1. The same modification is made whenever a packet is forwarded by a node. If, after decrementing the TTL, the TTL is 0, then the packet is silently dropped, rather than being rebroadcast. As specified by AODV, Node 2 creates a reverse-path entry in its Route Cache as shown below.

AODV Distance-Vector Cache (Node 2)				
Entry	Destination	Next Hop	Distance	Type
0	1	1	1	Reverse

AODV External Route Cache (Node 2)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

1.3 Both nodes 3 and 4 then each receive the **AODV_Route_Request**, process it, and respectively rebroadcast it since they cannot originate an **AODV_Route_Reply** message. Both nodes create a reverse-path entry in their Route Caches, as illustrated for Node 3 below.

AODV Distance Vector Cache (Node 3)				
Entry	Destination	Next Hop	Distance	Tpe
0	1	2	2	Reverse

AODV External Route Cache (Node 3)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

DSR Source-Route Cache (Node 3)		
Entry	Destination	Path
0	-	-

DSR External Route Cache (Node 3)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

As Node 3 is a Boundary node, operating both AODV and DSR, it passes the **AODV_Route_Request** to its Broker component which issues a **DSR_Route_Request**(`Target Address: 12, HopList: 3`) encapsulated in the same **Network_Layer**(`Source Address: 1, Destination Address: BROADCAST, TTL: 18, ID: 17, HopCount:2`) header subject to it clearing the rate limiting control of the External Requests Table. Since the Broker is issuing the **DSR_Route_Request** on behalf of another node it does not modify the Source Address in the Network-Layer header. Rather, to signify that Node 3 is the first hop in the DSR path, it adds Node 3 to the Hop List whereas heretofore DSR would have inferred that the source of the request was specified in the Network-Layer header's Source Address field. This protocol modification, which is a modification of an implementation detail rather than a modification of a technique, is carried out by all DSR nodes at all times. Consequently, DSR nodes expect the entire source-route, from first hop to last hop, to be specified in the Hop List.

The TTL of the **Network_Layer** would be shortened if the network diameter of the DSR region was less than the maximum initial TTL contained in the **Network_Layer** header received from the AODV network region, thus ensuring that discovered routes are viable in the DSR network region. The Source Address and Packet ID of the Network-Layer header are not modified during the process so that Node 1 is identifiable as the initiator of the request process at every stage of the process, in any part of the internetwork. Node 4 would also issue a **DSR_Route_Request** with the same Network-Layer header end-to-end field values, allowing DSR nodes to quench the storm of requests.

1.4 Node 5 does not operate the AODV protocol and cannot process or rebroadcast the **AODV_Route_Request** messages that it may receive from Node 3 and Node 4; it simply drops those messages. Node 5 processes the **DSR_Route_Request** message according to the DSR protocol, possibly caching the routing information in the message as shown below. Node 5 then rebroadcasts the **DSR_Route_Request**(Target Address: 12, Hop List: 3,5) encapsulated in the modified **Network_Layer** header, **Network_Layer**(Source Address: 1, Destination Address: BROADCAST, TTL: 17, ID: 17, HopCount:3). Since the Source Address in the **Network_Layer** header is not the same as the first hop in the Hop List, contained in the **DSR_Route_Request**, Node 5 may deduce that Node 1 is an External Node accessible through Node 3. It adds an entry to its External Route Cache for Node 1, noting that this node is accessible though Boundary Node 3, as shown below. Node 5 may calculate the distance from Boundary Node 3 to External Node 1 by subtracting the distance from it to Node 3, i.e. the original Hop List length, from the current Hop Count in the Network_Layer, giving an External Distance of 2.

DSR Source-Route Cache (Node 5)		
Entry	**Destination**	**Path**
0	3	3

DSR External Route Cache (Node 5)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	1	3	2

1.5 Node 7 processes the **DSR_Route_Request** normally, adding itself to the Hop List. It then broadcasts the **DSR_Route_Request**(Target Address: 12, HopList: 3,5,7) encapsulated in the modified **Network_Layer**(Source Address: 1, Destination Address: BROADCAST, TTL: 16, ID: 17, Hop Count: 4) packet. Node 7 may also cache the route data as Node 5 did.

1.6 Node 9 then receives the **DSR_Route_Request** from Node 7 and caches the routing information as Nodes 5 and 7 did, as shown below. Since Node 9 is not the target of the request and cannot answer the request from its Route Cache, it passes the request to its Broker component as it is a Boundary node operating both DSR and OLSR. Since the OLSR component of the node operates proactively, it will have a route to Node 12 in its link-state cache as the node is connected, as shown below.

DSR Source-Route Cache (Node 9)		
Entry	**Destination**	**Path**
0	3	7, 5, 3

DSR External Route Cache (Node 9)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	1	3	2

OLSR Link-State Cache (Node 9)				
Entry	**Destination**	**Next Hop**	**Distance**	**Boundary Node**
0	10	10	1	Yes
1	11	11	1	No
2	12	11	2	No

OLSR External Route Cache (Node 9)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

Since the length of the route to Node 12 is less than the **Network_Layer** TTL, the Node 9 Broker component can send a **DSR_Route_Reply** to Node 3 which is the first node in the Hop List, i.e. the source of the DSR part of the request, with the following header parameters; **DSR_Route_Reply**(Hop List: 3, 5, 7, 9, 12, Last Hop External: Yes, Distance: 2) encapsulated in a **Network_Layer** packet with the following header values, **Network_Layer**(Source Address: 9, Destination Address: 1, TTL: 20, ID: 23, Hop Count: 0). The reply is sent using the normal DSR mechanisms; it either uses a cached route, if the node has been able to cache routing information, or else the reply piggybacks a request for Node 3, as specified by DSR.

1.7 Normal DSR processes let the reply get to Node 3; each node caches routing information in accordance with the DSR configuration at each node. As the Destination Address of the **Network_Layer** header is 1, which is not the final address in the **DSR_Route_Reply** Hop List, the Broker component of Node 3 may deduce that a reply should be forwarded on to Node 1. The Broker issues an **AODV_Route_Reply** and **AODV_External_Network_Route_Reply** to Node 1. The **AODV_Route_Reply**(Hop Count: 2, Destination Address: 3, Destination Sequence Number: 2, Originator Address: 1) is sent as if the Destination Address requested in the original request message was Node 3. This allows Node 1 to build a forward path to Node 3. The **AODV_External_Network_Route_Reply**(Boundary Node: 3, External Node 12, External Distance: 5) allows the nodes along the path to Node 1 to learn of the route to External Node 12. These packets are encapsulated in the **Network_Layer**(Source Address: 9, Destination Address: 1, TTL: 17, ID: 23, Hop Count: 3) packet. The Broker Node specifies the **AODV_External_Network_Route_Reply** External Distance as being 5 as the distance is equal to the current Hop Count added to the Distance specified in the **DSR_Route_Reply**.

1.8 Nodes 1 and 2 process the **AODV_Route_Reply**(Hop Count: 2, Destination Address: 3, Destination Sequence Number: 2, Originator Address: 1) message as normal, creating a forward path to Node 3. They then process the **AODV_External_Network_Route_Reply**(Boundary Node: 3, External Node 12, External Distance: 5) by adding the information to their External Route Caches, as shown below for Node 1.

AODV Distance Vector Cache (Node 1)				
Entry	Destination	Next Hop	Distance	Type
0	3	2	2	Forward

AODV External Route Cache (Node 1)			
Entry	External Node	Boundary Node	External Distance
0	12	3	5

The route discovery phase for this scenario is completed once the reply message reaches the source node. Both the source node and the intermediate nodes that have cached the route information may now begin to switch packets on an end-to-end basis from the AODV region across to the OLSR region. The route maintenance mechanisms of the reactive regions begin to operate once the route has been established.

8.4.2 Route Discovery: Scenario Two

In the second scenario, as depicted at the top of Figure 8.13, a network is composed of three regions of nodes. A DSR region is adjacent to an OLSR region which is adjacent to an AODV region. Figure 8.13 also provides an illustration of the messages that are passed between the nodes in this example, corresponding to the description provided below. Node 1 tries to discover a route to node 12 for a data packet that has been passed to the network-layer from a higher layer. The DSR protocol places a copy of the data packet in its FIFO Send Buffer and then begins the route discovery process. None of the reactive nodes in this example have previously cached routing information.

2.1 Node 1 originates a route request, **`DSR_Route_Request`**`(Target Address: 12, Hop List:1)` subject to its own controls as set out in Appendix A, encapsulated in a Network Layer Packet with the following header values; **`Network_Layer`**`(Source Address: 1, Destination Address: BROADCAST,TTL: 20, ID: 13, HopCount:0)`. The Route Cache and External Route Cache for Node 1 are empty at this stage, as shown below.

DSR Source-Route Cache (Node 1)		
Entry	Destination	Path
0	-	-

DSR External Route Cache (Node 1)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

2.2 Node 2 receives the **`DSR_Route_Request`**, processes it, and rebroadcasts it since it cannot originate a **`DSR_Route_Reply`** itself. The Hop Count in both the request and the **`Network_Layer`** is incremented by 1 and the TTL of the **`Network_Layer`** header is decremented by 1. The same modification is made whenever a packet is forwarded by a node. If, after decrementing the TTL, the TTL is 0, then the packet is silently dropped, rather than being rebroadcast. Node 2 caches the route to Node 1 in its Route Cache as shown below.

DSR Source-Route Cache (Node 2)		
Entry	Destination	Path
0	1	1

DSR External Route Cache (Node 1)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

2.3 Nodes 3 and 4 are Boundary nodes operating both the OLSR and DSR protocols. As Nodes 9 and 10 provide access from an OLSR region to a non-OLSR region, they advertise their connectivity to an External Network by issuing OLSR_HNA messages in accordance with the OLSR specification. The OLSR_HNA messages are received and processed by the other nodes in the OLSR region. Nodes 3 and 4 would create entries in their Route Caches to record the existence of Nodes 9 and 10 as Boundary Nodes. The Route Cache for Node 1 is shown below.

DSR Source-Route Cache (Node 3)		
Entry	Destination	Path
0	3	7, 5, 3

DSR External Route Cache (Node 3)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

OLSR Link-State Cache (Node 3)				
Entry	Destination	Next Hop	Distance	Boundary Node
0	4	4	1	Yes
1	5	5	1	No
2	6	5	2	No
3	7	5	2	No
4	8	5	3	No
5	9	5	3	Yes
6	10	5	4	Yes

OLSR External Route Cache (Node 3)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

Both Node 3 and Node 4 would then independently send an **OLSR_HNAQ**(`External Node Address: 12, External Distance: 2`) message subject to the normal OLSR controls. These messages would be flooded through the OLSR network as far as Nodes 9 and 10. OLSR control messages are always originated in a new Network-Layer header as the header is used by receiving nodes to interpret OLSR routing information. As the messages flood the network they would be quenched according to the rate limiting control of the External Requests Table, as specified in Section 8.3.4.

2.4 Nodes 9 and 10 receive the **OLSR_HNAQ** messages encapsulated in **Network_Layer**(`Source Address: 3, Destination Address: BROADCAST, TTL: 18, ID: 14, Hop Count: 2`) packet. This message is passed to the Broker component of Boundary Node 9, which then issues an **AODV_Route_Request**(`Originator Address: 9, Originator Sequence No.: 1, Destination Address 12, Destination Sequence No.: Unknown, Hop Count: 0`). The encapsulating packet has the following modified values, **Network_Layer**(`Source Address: 4, Destination Address: BROADCAST, TTL: 20, ID: 17, Hop Count: 0`). Node 4 would also issue a **AODV_Route_Request** message which would have the same header values, allowing the DSR network region to quench the request storm.

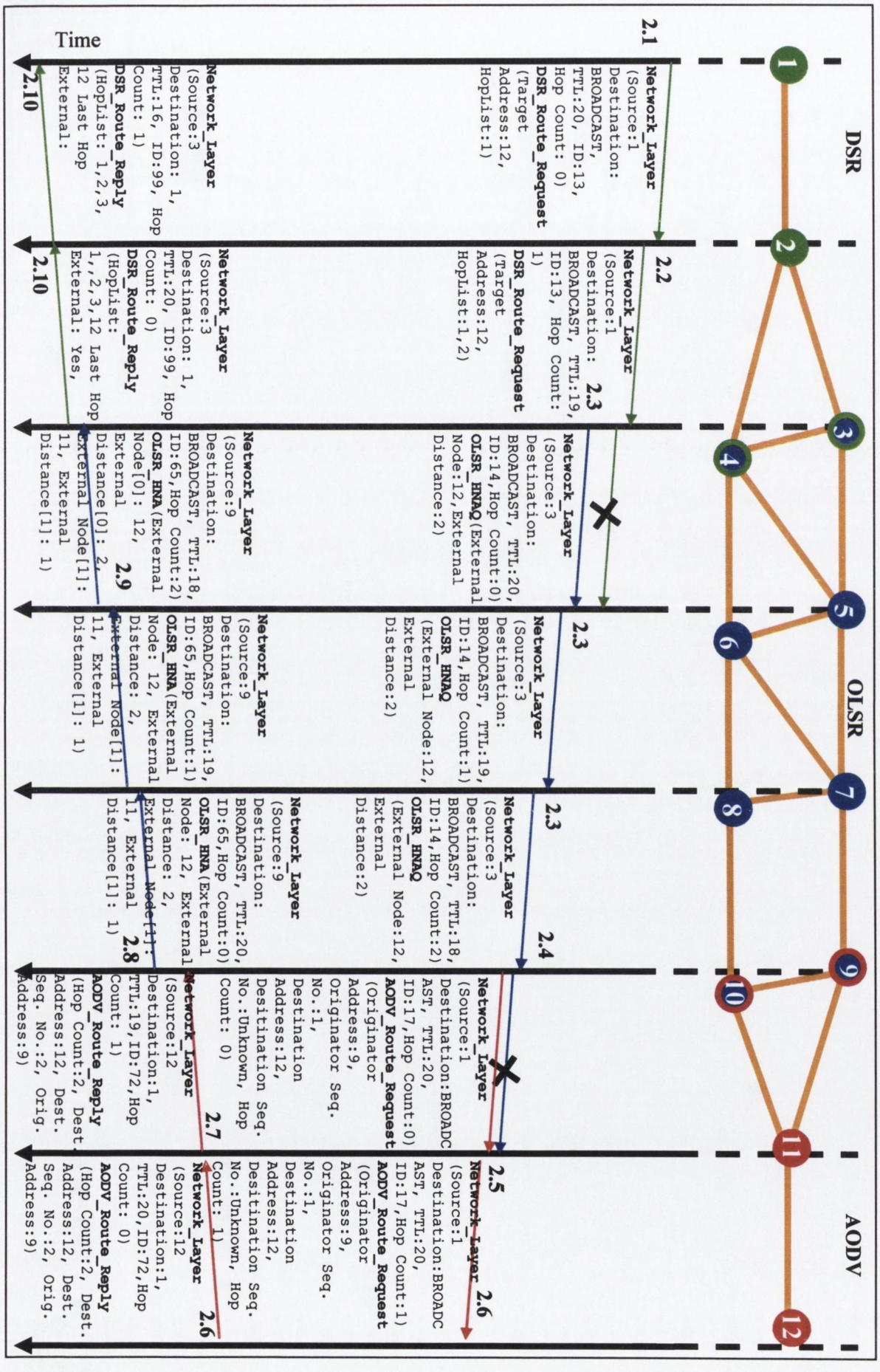


Figure 8-13 Scenario Two: Network Diagram and Route Discovery Messages

2.5 Node 11 processes the **AODV_Route_Request** as normal, creating a reverse-path to node 9 and rebroadcasting the packet.

AODV Distance Vector Cache (Node 11)				
Entry	**Destination**	**Next Hop**	**Distance**	**Type**
0	9	9	1	Reverse

AODV External Route Cache (Node 11)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

2.6 Node 12 receives the **AODV_Route_Request** for which it is the target. Node 12 originates an **AODV_Route_Reply**(Hop Count: 2, Destination Address: 12, Destination Sequence Number: 1, Originator Address: 9) message encapsulated in a **Network_Layer**(Source Address: 12, Destination: 9, TTL: 20, ID 72, Hop Count: 0) packet. The message is unicasted back to Node 9 along the reverse-path that was set-up by the request process.

AODV Distance Vector Cache (Node 12)				
Entry	**Destination**	**Next Hop**	**Distance**	**Type**
0	9	11	2	Reverse

AODV External Route Cache (Node 12)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

2.7 Node 11 processes the **AODV_Route_Reply** as normal, caching the forward-route information. The **Network_Layer** fields are adjusted so that the TTL is decremented and the Hop Count is decremented.

2.8 Node 9 then receives and processes the **AODV_Route_Reply** as normal, caching the forward-route information. Node 9 then issues **OLSR_HNA**(External Node[0]: 12, External Distance[0]: 2, External Node[1]: 11, External Distance[1]: 1) according to regular OLSR rules. The other OLSR nodes also cache the route information.

AODV Distance Vector Cache (Node 9)				
Entry	Destination	Next Hop	Distance	Type
0	11	11	1	Forward
1	12	11	2	Forward

AODV External Route Cache (Node 9)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

OLSR Link-State Cache (Node 9)				
Entry	Destination	Next Hop	Distance	Boundary Node
0	10	10	1	Yes
1	7	7	1	No
2	8	7	2	No
3	5	7	2	No
4	6	7	2	No
5	3	7	3	Yes
6	4	7	3	Yes

OLSR External Route Cache (Node 9)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

2.9 When Node 3 receives the **OLSR_HNA** it will cache the route information regarding the External Route to Node 12. Whenever a Boundary Node's OLSR Route Cache is updated it checks its External Requests Table to ascertain whether there is a match for a request for a route that is still pending. Node 3 finds a pending request from Node 1 with a Hop Count of 2. Since this Hop Count and the distance specified in the OLSR Route Cache are less than the original DSR TTL which is also specified in the External Requests Table, Node 3 issues a **DSR_Route_Reply(**`Hop List: 1, 2, 3, 12, Last Hop External: Yes, Distance: 5`**)** encapsulated in a new **Network_Layer** packet with the following header values, **Network_Layer(**`Source Address: 3, Destination Address: 1, TTL: 20, ID: 99, Hop Count: 0`**)**.

2.10 Nodes 1 and 2 process the reply, in which the Last Hop External Flag is set true, adding the External Route information to their External Route Caches.

The route discovery phase for this scenario is completed once the reply message reaches the source node. Both the source node and the intermediate nodes that have cached the route information may now begin to switch packets on an end-to-end basis from the DSR region across to the AODV region. The route maintenance mechanisms of the reactive regions begin to operate once the route has been established.

8.5 Inter-Regional Packet Forwarding and Route Maintenance

Once a node has been established across an ad hoc internetwork, packets may be switched from source to destination. However, as established routes may break down, the internetworking nodes must maintain routes and inform the relevant nodes and regions that the routes have failed.

These explanatory examples continue the network scenarios that have already been used to elucidate the Route Discovery techniques in the previous section. The worked examples demonstrate how the basic unreliable packet-switching service is provided across a heterogeneous internetwork. The route maintenance

technique is shown in tandem with the packet-switching technique as reactive route maintenance only occurs on active routes over which packets are being switched.

In a multi-region network, a route may break down on either side of a Boundary Node, i.e. in the Home Network or in the External Network. The maintenance of routes in a multi-region scenario reduces to informing the source node that a route has failed. If the route fails within the Home Network, then the normal protocol route maintenance procedures are followed. However, if the failure occurs in the External Network, then the error must be relayed back to the source node in the Home Network. In the following examples, the destination node fails as a packet is being forwarded towards it. Failure may occur if Node 12 shuts down or moves away from the node which connects it to the established route.

8.5.1 Packet Switching and Route Maintenance: Scenario One

The first scenario is depicted at the top of Figure 8.14. The scenario replicates that shown in Figure 8.12 in which a route from Node 1 to Node 12 was established. Figure 8.14 also depicts the message headers that are used during the packet-switching and route maintenance phases in this network scenario.

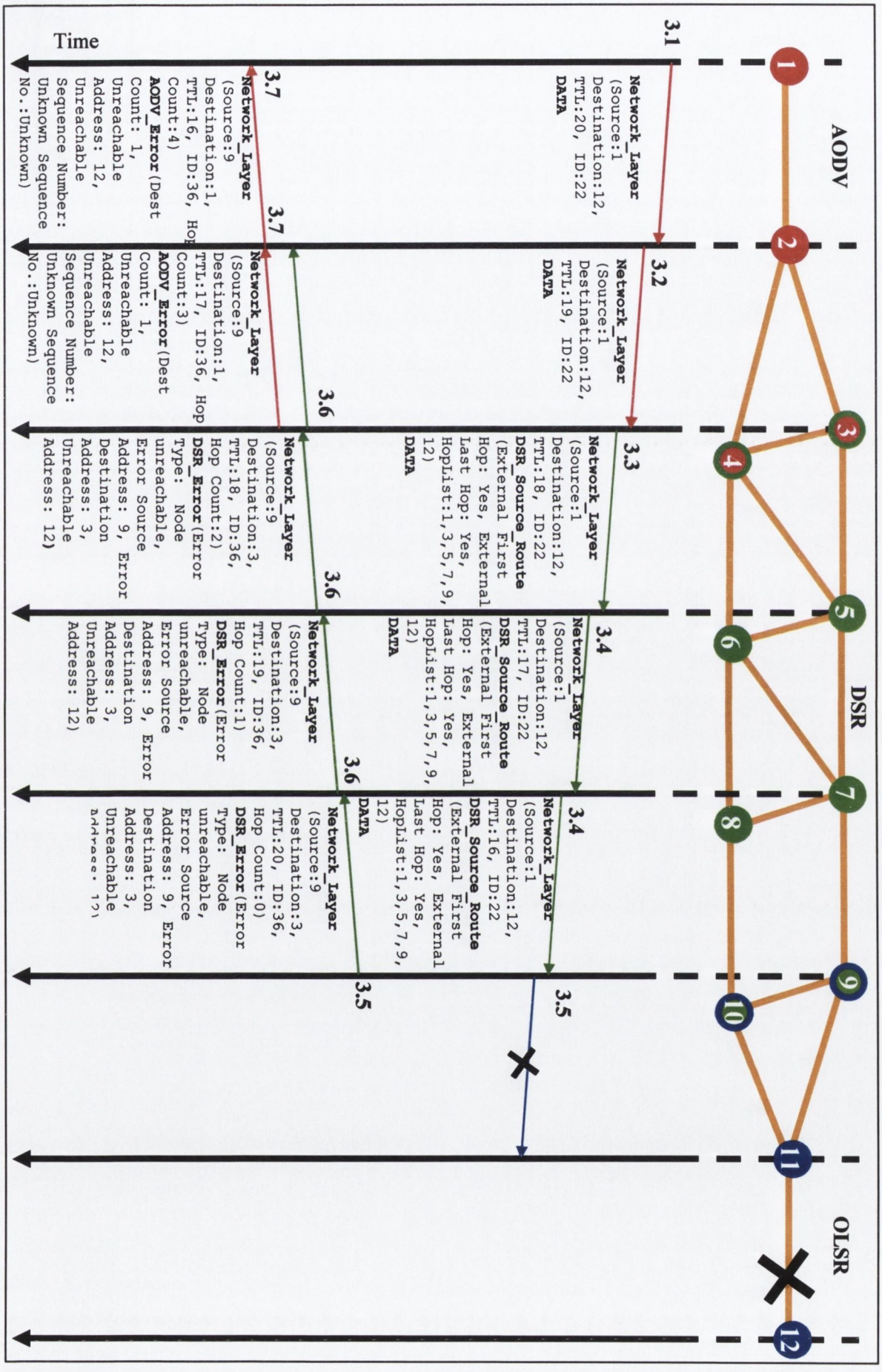


Figure 8-14 Scenario One: Network Diagram and Packet-Switching, Route Maintenance Messages

When a data packet is passed to the network-layer from a higher layer intended for Node 12, Node 1 attempts to switch the packet to Node 12 using the recently established route which it has cached.

3.1 Node 1 finds a route to Node 12 in its External Route Cache. This cache contains an entry for Node 12 specifying Node 3 as the Boundary Node that provides access to the External Network. Node 1 then successfully queries its Route Cache for a route to Node 3. Node 1 forwards the data packet to Node 2 encapsulated in a **`Network_Layer`** packet with the following header values, **`Network_Layer`**`(Source Address: 1, Destination 12, TTL: 20, ID: 22)`.

AODV Distance Vector Cache (Node 1)				
Entry	**Destination**	**Next Hop**	**Distance**	**Type**
0	3	2	2	Forward

AODV External Route Cache (Node 1)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	12	3	5

3.2 Node 2's packet-switching mechanism processes the packet as Node 1 did. It queries its Route Cache and External Route Cache for a route to 12 and then forwards the packet to Node 3, encapsulated in **`Network_layer`** packet with the following header values, **`Network_Layer`**`(Source Address: 1, Destination 12, TTL: 19, ID: 22)`.

3.3 Node 3 processes the packet as Nodes 1 and 2 did. It unsuccessfully queries its AODV Route Cache and AODV External Route Cache. The packet is passed to the Broker component as the Boundary Node operates AODV and DSR. The Broker successfully queries the DSR External Route which indicates that Node 12 is accessible through Boundary Node 9. The Node 3 Broker then successfully queries the DSR Route Cache for a route to Boundary Node 9. Node 3 source routes the packet to Node 9 by attaching a **`DSR_Route_Header`**`(First Hop External: yes, Last Hop External: Yes, Hop List: 1, 3, 5, 7, 9, 12)` to the packet and encapsulating it in the same **`Network_Layer`** packet with the following header values, **`Network_Layer`**`(Source Address: 1, Destination Address: 12, TTL: 18, ID:22, Hop Count: 2)`.

AODV Distance Vector Cache (Node 3)				
Entry	**Destination**	**Next Hop**	**Distance**	**Tpe**
0	1	2	2	Reverse

AODV External Route Cache (Node 3)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

DSR Source-Route Cache (Node 3)		
Entry	**Destination**	**Path**
0	9	5,7,9

DSR External Route Cache (Node 3)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	12	9	2

3.4 Nodes 5 and 7 process the source-routed packet as specified by DSR, decrementing the TTL and incrementing the Hop Count in the **`Network_Layer`** header as appropriate. Unlike the **`DSR_Route_request`** and **`DSR_Route_Reply`** messages, intermediate nodes cannot cache External Route information contained in the **`DSR_Route_Header`** message as it does not provide enough

information about the External Node; such nodes are identified as external to the DSR region by the First and Last Hop External flags. However, nodes may continue to cache route information about the DSR portion of the source route.

3.5 Node 9 receives the data packet as the last node specified in the **DSR_Route_Header**. However, as Node 9 is not the destination specified in the **DSR_Route_Header** Hop List field, it passes the packet to the Broker component. As Node 9 is a Boundary Node it checks its alternate OLSR Route Cache for a route to Node 12. However, as Node 12 has failed, normal OLSR techniques will have removed the link-state information about Node 12 from the OLSR Route Cache at Node 9. When the Broker component at Node 9 queries the OLSR Route Cache for a route to Node 12 the query will fail. The Broker component then issues a **DSR_Route_Error**(Error Type: Node unreachable, Error Source Address: 9, Error Destination Address: 3, Unreachable Address: 12) message specifying that Node 12 is unreachable. This message is unicast to Node 3, encapsulated in a **Network_Layer** packet with the following header values, **Network_Layer**(Source Address: 9, Destination Address 1, TTL: 20, ID: 36 Hop Count: 0). The Destination Address of the **Network_Layer** packet reflects the real source of the data packet, Node 1.

DSR Source-Route Cache (Node 9)		
Entry	**Destination**	**Path**
0	3	7, 5, 3

DSR External Route Cache (Node 9)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	1	3	2

OLSR Link-State Cache (Node 9)				
Entry	**Destination**	**Next Hop**	**Distance**	**Boundary Node**
0	11	11	1	No

OLSR External Route Cache (Node 9)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

3.6 The error message gets to Node 3 by means of normal DSR processing and forwarding techniques. Nodes 7, 5 and 3 remove their External Route Cache entries for Node 12. Since Node 3 is a Boundary node, the **DSR_Error**(Error Type: Node unreachable, Error Source Address: 9, Error Destination Address: 3, Unreachable Address: 12) is passed to the Broker component which then issues an **AODV_Error**(Dest Count: 1, Unreachable Address: 12, Unreachable Sequence Number: Unknown) encapsulated in the modified **Network_Layer**(Source Address: 9, Destination Address 1, TTL: 17, ID: 36 Hop Count: 3) packet. The Unreachable Sequence Number is specified as unknown as the node was not an AODV node within the AODV Home Network.

3.7 Nodes 1 and 2 process the **AODV_Error**(Dest Count: 1, Unreachable Address: 12, Unreachable Sequence Number: Unknown) message in the normal manner. External Route Cache entries for Node 12 are removed. As Node 12 is not an AODV node within Node 1 or Node 2's region, these nodes do not require a sequence number for Node 12 to avoid the occurrence of routing loops.

8.5.2 Packet Switching and Route Maintenance: Scenario Two

The second scenario is depicted at the top of Figure 8.15. The scenario replicates that shown in Figure 8.13 in which a route from Node 1 to Node 12 was established. Figure 8.15 also depicts the message headers that are used during the packet-switching and route maintenance phases in this network scenario.

When a data packet is passed to the network-layer from a higher layer intended for Node 12, Node 1 attempts to switch the packet to Node 12 using the recently established route which it has cached.

4.1 Node 1 finds a route to Node 12 in its External Route Cache, specifying Node 3 as the Boundary Node that provides access to the External Network. Node 1 then successfully queries its Route Cache, shown below, for a route to Node 3. Node 1 forwards the data packet to Node 2 encapsulated in a **`Network_Layer`** packet with the following header values, **`Network_Layer`**`(Source Address: 1, Destination 12, TTL: 20, ID: 41)` packet including a **`DSR_Route_Header`**`(First Hop External: No, Last Hop External: Yes, Hop List: 1, 2, 3,12)`. When Node 2 receives the packet it switches it to Node 3 in accordance with the **`DSR_Route_Header`** source route information.

DSR Source-Route Cache (Node 2)		
Entry	**Destination**	**Path**
0	3	2,3

DSR External Route Cache (Node 1)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	12	3	5

4.2 Node 3 processes the packet as Nodes 1 and 2 did. It unsuccessfully queries its DSR Route Cache and DSR External Route Cache. The packet is passed to the Broker component as the Boundary Node operates DSR and OLSR. The Broker successfully queries the OLSR Route Cache, as shown below, which indicates that Node 12 is accessible.

DSR Source-Route Cache (Node 3)		
Entry	**Destination**	**Path**
0	3	7, 5, 3

DSR External Route Cache (Node 3)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	-	-	-

OLSR Link-State Cache (Node 3)				
Entry	**Destination**	**Next Hop**	**Distance**	**Boundary Node**
0	4	4	1	Yes
1	5	5	1	No
2	6	5	2	No
3	7	5	2	No
4	8	5	3	No
5	9	5	3	Yes
6	10	5	4	Yes

OLSR External Route Cache (Node 3)			
Entry	**External Node**	**Boundary Node**	**External Distance**
0	12	9	2

The Node 3 Broker then forwards the packet into the OLSR region. The packet is switched to Node 10 through the OLSR region using the regular OLSR technique. Each intervening OLSR node will have stored routing information regarding Node 12.

4.3 Node 9 receives the data packet destined for Node 12. Node 9 unsuccessfully queries its OLSR Route Cache and it passes the packet to the Broker component. The Broker finds a route to Node 12 in the AODV Route Cache, shown below. The Broker component then forwards the packet to Node 11 with the following header values, **Network_Layer(**Source Address: 12, Destination Address 3, TTL: 20, ID: 55 Hop Count: 0**)**.

AODV Distance Vector Cache (Node 9)				
Entry	Destination	Next Hop	Distance	Type
0	11	11	1	Forward
1	12	11	2	Forward

AODV External Route Cache (Node 9)			
Entry	External Node	Boundary Node	External Distance
0	-	-	-

OLSR Link-State Cache (Node 9)				
Entry	Destination	Next Hop	Distance	Boundary Node
0	10	4	1	Yes
1	7	7	1	No
2	8	7	2	No
3	5	7	2	No
4	6	7	3	No
5	3	7	3	Yes
6	4	7	3	Yes

OLSR External Route Cache (Node 9)			
Entry	External Node	Boundary Node	External Distance
0	1	3	2

4.4 Node 11 attempts to switch the packet to Node 12 but discovers that the link to Node 12 has failed. Node 11 may attempt an AODV local repair on the route to Node 12. If that fails too, Node 11 then issues a **AODV_Error(**Dest Count: 1, Unreachable Address: 12, Unreachable Sequence Number: Unknown**)** message as normal encapsulated in a **Network_Layer(**Source Address: 1, Destination 12, TTL: 20, ID: 54**)** packet.

4.5 The **AODV_Error(**Dest Count: 1, Unreachable Address: 12, Unreachable Sequence Number: Unknown**)** message gets to Node 9 by means of normal AODV techniques. The entry in Node 12's Route Cache for Node 9 is removed. Node 9 no longer issues **OLSR_HNA** messages advertising Node 12.

4.6 At Nodes 4 and 6 the route information regarding Node 12 is removed from the OLSR Route Cache as it is not refreshed by the receipt of **OLSR_HNA** messages. Intermediate nodes in the OLSR region also remove entries for Node 12 as per protocol-specific mechanisms.

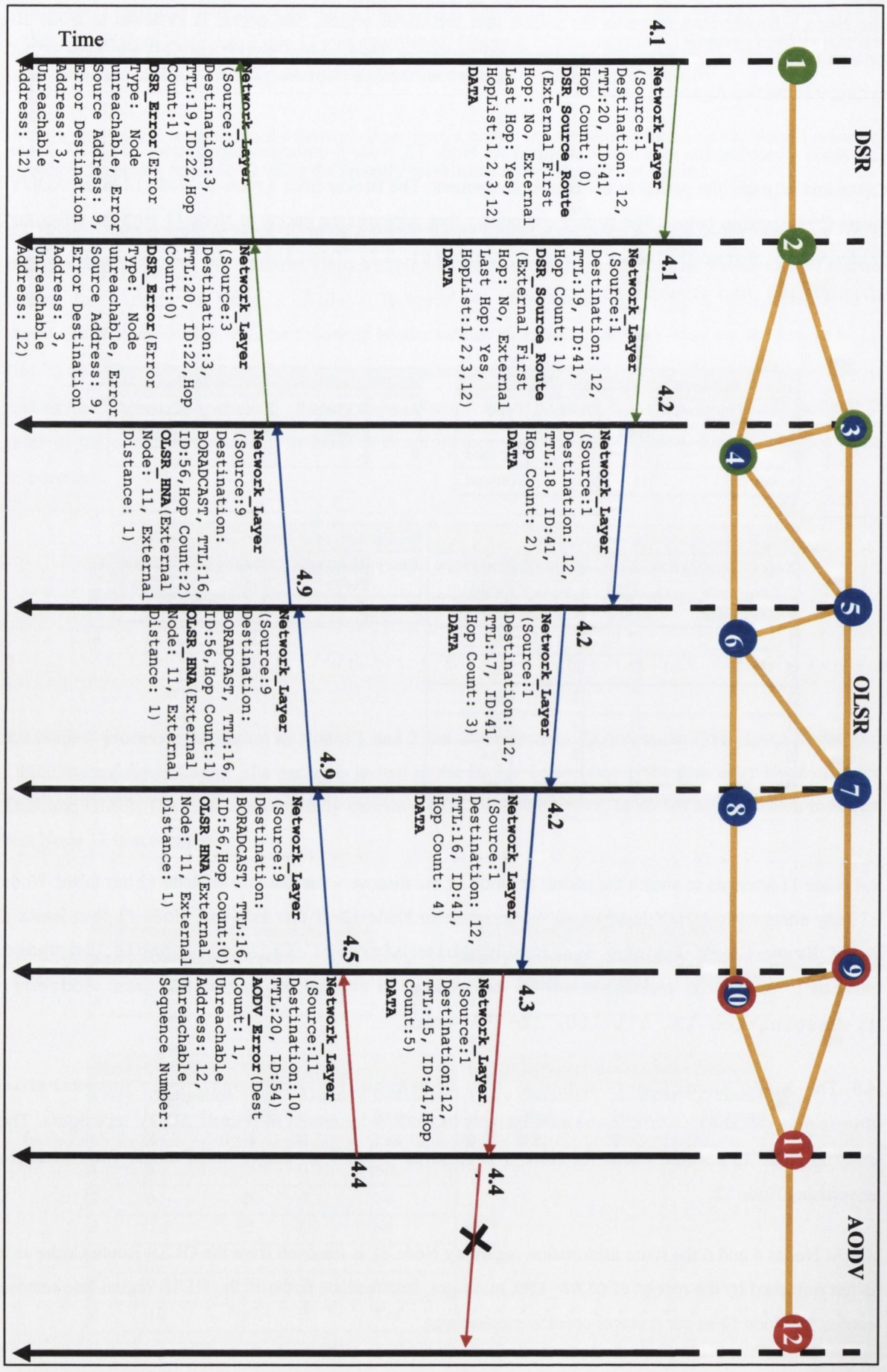


Figure 8-15 Scenario One: Network Diagram and Packet-Switching, Route Maintenance Messages

4.7 Upon the receipt of the next data packet from Node 1 destined to Node 12, Node 3's query of its route caches will fail to find a route to Node 12. Node 3 then issues a **DSR_Error**(Error Type: Node unreachable, Error Source Address: 3, Error Destination Address: 1, Unreachable Address: 12).

4.8 Otherwise, the **DSR_Error**(Error Type: Node unreachable, Error Source Address: 9, Error Destination Address: 3, Unreachable Address: 12) is processed by Node 1 and Node 2 which remove Node 12 from their respective External Route Caches. Both nodes may then try to re-establish a fresh route if there is a demand.

8.6 Summary

This chapter has focussed on issues pertaining to the ability of a heterogeneously configured ad hoc network to allow ad hoc internetworking between various regions that constitute the entire network. Ad hoc internetworking has been identified as one of the key features that enable the creation of networks that may react to network demands by reconfiguring regions of the network to use different ad hoc routing protocols which are suited to the underlying network conditions.

It is readily apparent that the existing concepts which are both used and proposed to enable internetworking between autonomous systems or between the Internet and a mobile ad hoc network would not suffice. The internetworking of heterogeneously configured conventional networks is based on the use of statically configured gateways and hierarchical backbone solutions which are completely unsuitable for a mobile wireless ad hoc network.

This chapter presented a comprehensive internetworking technique that enables connected regions of nodes to discover routes that extend from one region into another in a completely ad hoc, distributed manner that is as scalable as the underlying ad hoc routing protocols themselves. The concepts of Home Networks, External Networks, External Routes and External Route Caches were also introduced in this chapter. In order to maintain the scalability and efficiency of the constituent ad hoc routing protocols; AODV, DSR and OLSR, the internetworking technique uses the existing signalling techniques of each routing protocol to effect its business. This enables each of the protocols to engage in an ad hoc internetwork without changing the characteristics of the protocols or the robustness of the distributed system.

A Broker component was presented and its function was explained. The Broker sits in each Boundary Node and is responsible for ensuring that the scalability of the internetwork is maintained by throttling the rate at which control packets pass from one network into another. The Broker component is also responsible for transforming packets as they pass from one region into another so that they may be processed in the receiving region.

9 Conclusions

9.1 Introduction

This chapter presents the conclusions that may be drawn from this thesis, the essence of which is that it is both necessary and feasible to design a flexible system at the network-layer of an open, mobile, wireless, ad hoc network. Section 9.2 reviews the specific contributions identified in the introductory chapter and details how they were achieved. Section 9.3 highlights some future work that can follow on from this body of work. Section 9.4 concludes the thesis.

9.2 Review of Contributions

The main purpose of this work has been to demonstrate the necessity and feasibility of a robust flexible multi-protocol network-layer for use in open, dynamic and mobile wireless ad hoc networks. The following discussion reviews the contributions that have been made to substantiate that claim.

A comprehensive survey and analysis of ad hoc networking was presented, focussing extensively on the routing protocols used in such networks. A strong motivation for the design of a reconfigurable, multi-protocol network-layer was advanced.

A comprehensive overview of ad hoc networking was provided in this thesis, the main discussion of which was presented in Chapter 2. The review focussed on the diversity of form and function which characterises the technologies underlying the creation of ad hoc networking systems. The analysis revealed that there is much uncertainty as to the future form of ad hoc networks, both with regard to their purpose and also with regard to which core technologies will succeed. The review has shown that many new techniques are

proposed for use in ad hoc networks, replacing existing protocols and technologies which have been developed for static, planned and wired networks. It was shown that the mobile ad hoc networking domain demands that such communication stack technologies be re-examined and redesigned, not merely incrementally modified or tweaked for this challenging networking environment.

In an extensive analysis of multi-hop routing protocols it was shown that there is a wide array of multi-hop distributed routing protocols on offer, each of which have been shown in the literature to exhibit varying performance profiles when challenged by different networking scenarios. Consequently, a very detailed analysis of the causes of these variations in the performance of the routing protocols was carried out and the three networking factors of node density, relative node mobility and traffic loading were identified as having the greatest effect on an ad hoc node's ability to efficiently engage its routing protocol.

In sum, it was shown that ad hoc networks demand solutions tailored to their unique characteristics; the solutions and architectures associated with static, Internet-type systems will not suffice. The review concluded by proposing the development of a flexible multi-protocol network-layer that can independently and dynamically respond to the network's needs.

The concept of flexibility and adaptability in ad hoc networks was developed through the re-engineering of the structure of the network-layer for mobile ad hoc networks. The specific requirements of such a network-layer were reasoned and enumerated.

As the central aim of this thesis was to demonstrate the feasibility of the development of a flexible multi-protocol network-layer, extensive consideration was given to the development of a system that would support such a complex and sophisticated distributed system. The conclusion of this analysis was that system components enabling *auto-configuration* (initialisation, joining, migration and merging), *ad hoc internetworking* between heterogeneous regions of the network and network-layer *reconfiguration* of regions of nodes in response to changes in the observed networking conditions were key to maximising flexibility at the network-layer.

A consequential conclusion of that analysis was that current network-layer structures would not suffice for such an adaptive concept. Current network-layer structures are built with a changeover timeframe of years in mind whereas the flexible network-layer would be expected to organise itself over a timeframe of the order of seconds, such is the dynamic nature of the environment.

The analysis determined that those core features of a flexible network-layer centred on the network's ability to observe and analyse its environment with a view to either correcting or optimising its configuration in response. Notably, it was also determined that the components enabling such flexibility at the network-layer should not undermine the already-optimised ad hoc routing protocols in use at the network-layer. To that end, the explicit use of multi-hop routing services was eschewed throughout the design of the system.

A set of metrics for observing and analysing the networking conditions pertaining to the efficiency of a node's network-layer configuration was introduced. Techniques to enable the local and passive collation of such metrics were developed.

Flexibility to reconfigure the network-layer in this system assumes the ability to observe, analyse, reason and conclude about the configuration changes desired. It was shown that nodes must be enabled to appreciate the network factors that affect their operating efficiency in order to intelligently reconfigure the network. In order to observe and analyse such network factors, it was found necessary to develop metrics that fit with the system objectives. It was shown that there is some difficulty in determining network analysis metrics for autonomous nodes in a completely distributed, non-hierarchical system. The use of conventional techniques such as those used in network management schemes was rejected as such systems sit on top of the network-layer. Instead, costless, passive, localised metrics were devised, allowing each node to collate information that is freely available within the network-layer of each autonomous node. These metrics enable each node to independently measure the relative mobility of the node, the degree of the node and the traffic loading on the node.

Novel techniques for enabling the auto-configuration of ad hoc nodes were developed. These techniques ensure the stable and consistent configuration of a distributed multi-state networking system by providing for the robust and lightweight just-in-time correction of distributed network variables.

This mechanism was based on the requirement that all nodes operating a flexible multi-protocol network-layer should be able to independently manage their own network-layer configuration. The component had to be designed in a completely distributed manner, capable of dealing with any manner of naturally occurring disruptive network events. In designing the component, ad hoc management schemes, address auto-configuration schemes and the concept of self-stabilisation were comprehensively analysed and reviewed.

The concept of self-stabilization by local checking and correction was identified in this thesis as a suitable technique on which to base such a robust and distributed autonomous-configuration protocol. The resulting auto-configuration protocol was built on an initial tentative stateless network-layer configuration protocol (the SCI protocol) coupled with a non-terminating, self-stabilising protocol (the CCDR protocol) which detects and corrects inconsistencies in the network state.

Novel techniques for enabling the reconfiguration of ad hoc nodes were developed. These techniques enable a mobile ad hoc network to reconfigure its variable parameters in response to the networking factors that affect its performance efficiency.

This novel mechanism was motivated by the need for nodes to change their choice of routing protocol at runtime in response to influential changed network factors. The literature was shown to demonstrate that there are limits to which classical distributed consensus-type protocols could be applied to the mobile wireless ad hoc network setting on account of the Fischer-Lynch-Patterson (FLP) impossibility result. The novel technique presented in this thesis accepted those limits by its use of local, soft-state decision-making protocols which fall within the FLP constraints.

A novel framework was presented, one that builds on the social-science theory of diffusion of innovations, a theory which describes a flexible decision-making model. The use of this concept enabled the development

of an autonomous reconfiguration protocol which addresses the constraints imposed by the mobile ad hoc networking system model.

Novel techniques for enabling the ad hoc internetworking of heterogeneously configured ad hoc domains were developed. These techniques facilitate end-to-end, multi-hop, peer-to-peer networking in an internetwork of dynamically reconfigurable heterogeneous routing domains.

A novel technique was developed to enable the network to offer seamless multi-hop peer-to-peer ad hoc internetworking across a network composed of multiple heterogeneously configured regions (or domains). As shown in this thesis, the spatially and temporally diverse nature of open ad hoc networks strongly suggests the development of a more sophisticated multi-protocol network-layer which would enable the creation of an ad hoc internetwork of autonomous domains: each domain using an interior routing protocol which is optimally suited to the prevailing demands. Such a reconfigurable networking system is made possible by the above-cited novel techniques.

However, maintaining multi-hop, peer-to-peer connectivity in such a heterogeneous internetwork of autonomous and dynamically configured systems was shown to be a non-trivial task. An extensible framework was presented, one that builds on the existing proven and robust interior ad hoc routing protocols, such as AODV, DSR and OLSR, extending their design philosophies to the task of seamless ad hoc interdomain routing. The design of the framework took a *completely* distributed approach, in keeping with the robust design philosophy of the underlying interior routing protocols. The concepts of Home Networks, External Networks and External Route Caches, as well as techniques that enable nodes to discover and use routes across an ad hoc internetwork, were introduced in this work.

9.3 Future Work

This thesis has focussed on enabling ad hoc networks to autonomously organise themselves at the network-layer in response to the divergent networking demands and conditions that may be experienced in a realistically characterised ad hoc network. The contributions presented in this thesis have demonstrated that it is both necessary and feasible to look beyond contemporary network architectures and solutions when seeking to explore a domain as challenging as that represented by mobile ad hoc networks. However, the problems addressed in this thesis only represent a subset of the issues which must be addressed if ad hoc networks, and indeed any future packet-based wired or wireless networks, are to succeed in more divergent, more dynamic and more demanding environments than have ever existed heretofore. It has already been acknowledged in Chapter 7 that the whole issue of decision-making in complex distributed systems needs to be further advanced, possibly by trying to take more cues from the literature emanating from diverse disciplines such as economics and social science. As this thesis is an exploration of flexibility in ad hoc networks, a number of topics are suggested as meriting research in the near future.

9.3.1 Disaggregated Network Architectures

As technologies such as ad hoc networks become a reality, facilitating a move towards pure packet-based wired, wireless and hybrid networks, the global networking landscape will become increasingly fractured and heterogeneous in its makeup. As networks head in this direction, there will be a need to architecturally accommodate the increased flexibility required by increasingly disaggregated networks. Disaggregation is a term that is used to describe networks that are non-homogeneous, distributed and subject to control by competing entities. Network disaggregation can occur in any of the traditional layers of a network protocol stack or it may occur outside the technological sphere, i.e. in the commercial or regulatory spheres. Furthermore, network disaggregation may occur over both the space and time continua.

To further explain the concept of disaggregation, the notion of a control plane is introduced. Such a plane may exist for many purposes: *technical* – network route management, MAC management, distributed PKI management, resource discovery management; *commercial* – billing, ownership; or *regulatory* – RF spectrum licensing, anti-monopoly controls. The purpose of a control plane, regardless of what type it is, is to ensure a degree of organisation across an inherently distributed system. In the absence of such control planes, individual network entities cannot interact in a collaborative manner.

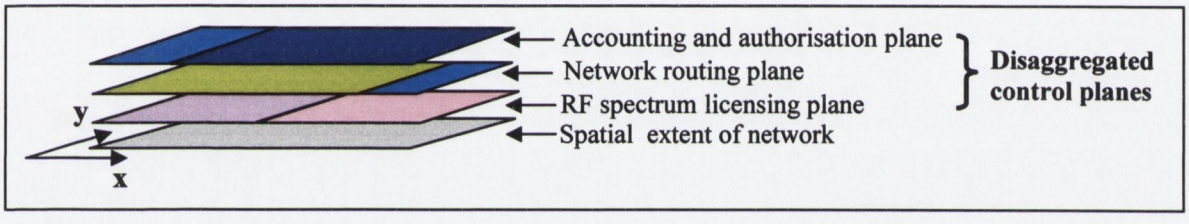


Figure 9.1 Control Plane Disaggregation over Space

As the scope of the network extends over both space and time, there will be a need to disaggregate the control within each plane so as to address local, i.e. time-local and space-local, conditions. Referring to Figure 9.1, it may be seen that there are three space-oriented control planes extending over this sample network. Looking at the *accounting and authorisation* plane, it may be seen that the plane is divided into two domains (the light and dark blue parallelograms). This infers that the network is subject to control by two authorities which have jurisdiction in two separate spatial locations. The *network routing* plane, arbitrarily positioned below the *accounting and authorisation* plane in Figure 9.1, is also split into two domains. However, it should be observed that the discontinuity in the control of the *networking routing* plane does not match that of the *accounting and authorisation* plane. In short, the decisions to disaggregate these two planes are independent events.

In Figure 9.2, the time-oriented disaggregation of the RF spectrum plane is illustrated, with a discontinuity occurring at a around noon. This disaggregation in RF spectrum control may represent a change in spectrum usage permissions at this time. Such a temporal change in control over RF spectrum may occur where service providers are given the right to use spectrum in accordance with user demand, e.g. spectrum may be given over to traffic information monitoring and dissemination in the morning to accommodate the heavy load of the rush hour and then transferred to environmental monitoring in afternoon for pollution and weather tracking services when those needs become dominant.

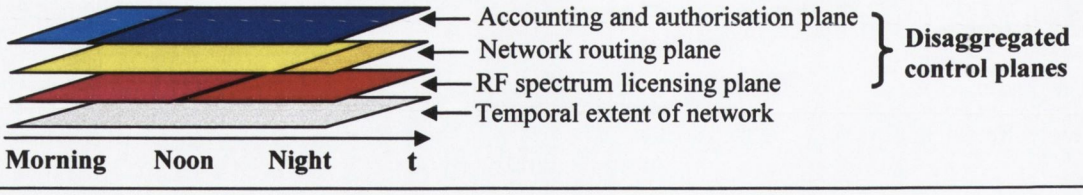


Figure 9.2 Control Plane Disaggregation over Time

Furthermore, while a control plane may map directly to a protocol layer, as in the case of the *network routing* plane, in other cases it may not. In such cases, the manner in which a node or network partition intersects the control plane may demand a new and different way of representing and implementing the technical management services and protocols necessary to enable disaggregation. Control planes may be independent of each other, as in the case of a *network routing* plane and an *application accounting* plane, or they may be dependent planes, e.g. the *ownership* plane and the *security* plane.

Relating this idea back to the work presented in this thesis, the network-layer routing protocols and packet-switching services represent an implementation of a control plane. As illustrated in both Figure 9.1 and Figure 9.2, the *network routing* plane is not uniformly controlled over all space and time. The techniques developed in this thesis represent the kinds of management services that are necessary to allow the network to disaggregate at the network-layer in order to accommodate diverse demands and conditions, whilst also ensuring that the end-to-end network communication services are still preserved.

This thesis demonstrates there are a number of concrete technical tasks that must be addressed before any control plane may be disaggregated: namely zero-configuration, collective reconfiguration and inter-domain transacting. Autonomous (or zero) configuration will always be necessary for any control plane where there can be no prior expectation of the state of the control plane prior to the node/network engaging with it. Collective reconfiguration techniques will be necessary for nodes/network partitions to cope with dynamic changes in the control plane. Finally, as the essence of a disaggregated network architecture is that it allows competing or alternative networks and network services to operate, there must be a facility to allow nodes communicate across a disaggregated global network in seamless manner.

9.3.2 Volatile Systems

In moving away from centralised, wired and static networks, to a network architecture that is much more fluid there will be a need to address the inherent volatility in such systems. Volatility is a term that may be used to describe the characteristics of a system that make it susceptible to uncontrollable, unforeseeable or unplanned changes which may have the effect of disrupting the structure of the networking system, e.g. the relationships between neighbouring, peer and end nodes. Any network that exhibits some of the following attributes may be seen as a system susceptible to volatility: *node mobility*, *open network structures*, *variable link quality*, *variable signalling techniques* and *variable connectivity* among others. The consequences of such volatility are that it disrupts distributed protocols, programmes and techniques which are heavily dependent on the underlying physical and topological structure of the network.

The structure of any network revolves around the nodes of that network and the links intervening between those nodes. When the network is wired and centralised, the links are generally reliable and the location of the nodes is not subject to unplanned change. The quality of the links is also assured to a high degree, such that centralised control and monitoring of the network state is possible. Such stability generally negates any need for dynamic reconfiguration within the system. Generally, reconfigurability is only desirable as a means to address some deficiency in the current network configuration; deficiencies brought about by changes in the networking conditions and demands. As networks become increasingly mobile and dependent on wireless links, the variability of the network's quality-of-service, i.e. its end-to-end throughput, delay and capacity characteristics, becomes larger. Such variability demands the reconfiguration of protocol and system parameters so that the network is best-configured with regard to the prevailing conditions.

Emerging network architectures, of which ad hoc networks are but one example, will be increasingly built around adaptable and reconfigurable hardware and software. In practice, such reconfigurability introduces the notion of multi-state systems that can change their state, or have it changed for them, in response to different needs. In the context of reconfigurable services and applications designed for distributed networking systems, the state of those services will often be determined by the union of the states at the individual nodes and links that make up that system.

In this thesis, issues of reconfigurability, multi-state protocols and system volatility were addressed by means of the auto-configuration protocol described in Chapter 6. However, such protocols and techniques must become more prevalent in all future networks susceptible to volatility, even volatility of the weakest kind such as a single failing link or link of inconsistent quality. In the absence of strong (in the distributed computing sense) leadership and monitoring services, individual nodes are responsible for ensuring the correction of their own state. To this end, the concepts that were introduced by Dijkstra [Dij74] must be applied more consistently to future protocols and distributed techniques designed for systems that may be easily perturbed.

9.3.3 Service-Driven Protocol Stacks

This thesis espoused the view that a reconfigurable network-layer facilitated the kind of flexibility that is necessary to address the challenges faced by realistically characterised ad hoc networks. In doing so, it was shown that there is a need to move from the static configuration of a node's network-layer to a more flexible system that allows nodes to configure themselves in response to changing networking conditions and demands. However, in this thesis, this service-driven reconfigurability was presented within the confines of statically formed protocol stacks that generally conform to the model of the TCP/IP or OSI-based protocol stacks. Such stacks are generally configured to have about seven layers ranging from the physical layer up to the application layer, regardless of specific demands that will be placed on those stacks. The whole concept of peer-to-peer networking, whether it is the interaction between clients and servers at the application-layer or remote nodes at the network-layer, is predicated on the understanding that all participating peer nodes will be using the same protocol stack.

In the context of open mobile ad hoc networking, it is assumed that each node entering a network has been setup with the same protocol stack structure. This ensures that all nodes in the network can interact on a

peer-to-peer basis, whether it is a neighbour to neighbour interaction at the physical-layer or an end-to-end transaction at the application-layer. This assumption could easily be argued to be a contrivance in the context of an open, autonomous computing environment. In much the same way that this thesis proposed that nodes should not be expected to have configured their network-layer *a priori*, the same case can be made for the construction of a node's protocol stack. Future networking systems will be far more distributed than today, depending on autonomous nodes and decentralised authority structures. There will be no standard node model to conform to as there will be no single regulatory network authority. The challenge will be to allow nodes to construct the appropriate protocol stack *on-the-fly* so as to address the current network demands. This type of dynamic stack construction will have to ensure consistency between neighbouring, peer and end-system nodes on a demand-driven basis.

Therefore, in moving towards an increasingly flexible and reconfigurable network architecture, a rethinking of the notion of statically deployed protocol stacks may be required. Rather than creating a network of identikit nodes in the absence of network experience, it may be possible to move to a service-driven protocol stack model in which a node creates (and destroys) its protocol stack in response to demand. Returning to the core of the protocol stack ideals, which were highlighted in Chapter 4, a well-formed protocol stack consists of layers with clearly defined service interfaces. If such an ideal is adhered to, then it should be possible to construct appropriate protocol stacks in an on-demand basis if techniques and protocols are developed to ensure that the necessary peer-to-peer compatibility of transacting nodes, which fundamental to the operation of any protocol stack concept, are respected.

9.3.4 Cost-Benefit Comparison of Competing Network Solutions

In order for emerging network architectures and technologies to advance, they must be adopted by the markets that they are targeted to. While the adoption of new technologies is influenced by an array of factors, some of which may be intangible (*the diffusion of the innovations*) or incalculable, the perceived cost-benefit ratio of a new technology will almost certainly influence its possible adoption. In order to enable the ready comparison of emerging architectures, it shall be necessary to objectively characterise both the costs and the benefits that accrue with each solution or architecture. However, as the set of solutions that will be available in the future will be quite diverse in terms of how they go about providing end-to-end data services, it will often be a case of comparing *apples* with *oranges*, so to speak. In order to compare the immediately incomparable, it will be necessary to reduce the essence of each solution to a $/kbps or $/QoS number so that the solutions are readily comparable for the potential user or adopter. In giving this number, it will also be equally important to ensure that the process of arriving at that number does not incur the same effort as the development of the networking solution in question.

For example, a simple target market scenario may be a college campus that requires the availability of data services throughout the campus. While it is accepted that the solution will entail wireless technologies, it is not clear as to what wireless solution should be deployed as many candidate solutions exist. One solution, the *apple* solution, may be an advanced 3G offering which would require a contract with the 3G network service provider, the installation of base-stations and the trenching of cables to the 3G Radio Network Controller. Another sample solution, the *orange* solution, may be an IEEE 802.16 wireless mesh network which may

connect to the college's existing wired LAN and require little or no cable and trenching. Neither would this solution require a new contract with an external network service provider. Such a scenario may present difficulties to the potential adopter as there are many cost-benefit factors to bear in mind: the initial capex, the ongoing opex, the expected QoS, the labour force costs (industrial relations, redundancies, new hires) and other intangibles (political cost, student-college relations) etc. A successful model should allow for an array of such diverse costs and benefits to be represented by as few simple metrics as possible, such as diverse motor cars are often represented and classified by their engine capacity and fuel efficiency.

In order to develop a model that can be used to provide a fast and accurate cost-benefit analysis of any given network architecture or solution, the use of a component-based model may be beneficial as it may enable reuse of components that appear in a number of solutions. For example, if it is possible to model the cost-benefit attributes of a wireless link subsystem or set of dependent link subsystems (the smallest component that may be modelled may not always have a direct correlation to the smallest network element as technical constraints may demand the grouping or coupling or multiple sub-components) then it may be easier to analyse the cost-benefit of larger systems that are built around these subsystems.

The real challenges will arise when technologies move towards increasingly flexible platforms, platforms which are increasingly reliant on reconfigurable software and simple, but fast and powerful, general purpose processing abilities. In such a scenario, the existing network may be able to entirely reconfigure itself to offer a new type of solution using the existing hardware, e.g. radio front ends, memory and processors, but new software, e.g. new modulation schemes, new MAC schemes and new network schemes. In this case, the change could occur much more rapidly than is currently the case of the installation of a network solution at a greenfield site. In essence, the change over to a new network solution over a relatively short period of time at the site of an exiting solution approximates to a brownfield site deployment, albeit at a much more rapid pace. The notion of this kind of brownfield-site system-upgrading will become much less discrete and obvious event in the future. In such network settings, rapid local or regional reconfiguration may become the norm. However, the same demands on cost-benefit estimation hold true in such a scenario as a network will only be reconfigured if there is a cost-benefit in doing so.

9.4 Conclusion

This thesis represents a step forward in our understanding of what flexibility, adaptability and reconfigurability means at the network-layer of mobile, wireless, ad hoc networks. Rather than focussing on developing *yet another* new routing protocol that would address the inconsistencies of the existing body of ad hoc routing protocols, this thesis has proposed a novel system that leverages the strengths of existing ad hoc routing protocols by enabling the network's nodes to autonomously organise themselves in response to changing networking conditions and demands. This work provides a sound basis on which to advance further research for future reconfigurable networking systems and to develop the services necessary for the successful deployment of next-generation wireless networking systems.

10 Bibliography

[Aid94] S. Aidarous and T. Plevyak (Editors), "Telecommunications Network Management into the 21st Century", *IEEE Press*, New York, NY, U.S.A., 1994.

[Ala91] Cengiz Alaettinoğlu, Klaudia Dussa-Zieger and Ibrahim Matta, "MaRS(Maryland Routing Simulator) – Version 1.0 User's Manual", *Technical Document, Department of Computer Science, University of Maryland, Maryland, U.S.A.*

[Aro94] Anish Arora and Mohamed Gouda, "Distributed Reset", *IEEE Transactions on Computers*, Volume: 43, Issue: 9 , Sept. 1994, Pages:1026 – 1038.

[Att98] Hagit Attiya and Jennifer Welch, "Distributed Computing: Fundamentals, Simulations and Advanced Topics", *McGraw-Hill Publishing*, England, 1998.

[Awe91] Baruch Awerbuch, Boaz Patt-Shamir and George Varghese, "Self-Stabilization By Local Checking and Correction", *Proceedings of the 32nd IEEE Annual Symposium on Foundations of Computer Science*, 1-4 Oct. 1991, Pages: 268 – 277.

[Bah02] Venkat Bahl, "Zigbee and Bluetooth – Competition or Complementary?", *Zigbee Alliance Presentation*, September 2002. http://www.zigbee.org.

[Bak95] F. Baker, "Requirements for IP Version 4 Routers", *IETF RFC 1812*, June 1995. http://www.ietf.org

[Bas98] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, Barry A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)", *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Dallas, Texas, 1998, Pages: 76 – 84.

[Bas99] S. Basagni, I. Chlamtac and V.R. Syrotiuk, "Dynamic source routing for ad hoc networks using the global positioning system", In the *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 21-24 Sept. 1999, Pages: 301 -305.

[Beu03] Jan Beutel, Oliver Kasten, Matthias Ringwald, Frank Siegemund and Lothar Thiele, "Bluetooth Smart Nodes for Mobile Ad-hoc Networks", *Technical Report No. 167*, Eidgenössische Technische Hochschule (ETH) Zürich, Switzerland, April 2003.

[Bey90] D. Beyer, "Accomplishments of the DARPA SURAN Program", *Military Communications Conference*, 1990. MILCOM '90, Conference Record, 'A New Era'. 1990 IEEE , 30 Sept.-3 Oct. 1990, Pages: 855 -862.

[Bha94] Vaduvar Bharghavan, Alan Demers, Scott Shenker and Lixia Zhang, "MACAW: A Media Access Protocol for Wireless LANs", *ACM SIGCOMM Computer Communication Review* , Proceedings of the conference on Communications, Architectures, Protocols and Applications, Volume 24 Issue 4, 1994, Pages: 212 – 225.

[Bha99] Pravin Bhagwat and Adrian Segall, "A Routing vector Method (RVM) for Routing in Bluetooth Scatternets", *IEEE International Workshop on Mobile Multimedia Communications*, 1999. (MoMuC '99), 15-17 Nov. 1999, Pages: 375 -379.

[Bla01] Ljubica Blažević, Levente Buttyán, Srđan Čapkun, Silvia Giordano, Jean-Pierre Hubaux and Jean-Yves Le Boudec, "Self-Organization in Mobile Ad-Hoc Networks: the Approach of Terminodes", *IEEE Communications Magazine*, Volume: 39 Issue: 6, June 2001, Pages: 166 -174.

[Bla02] John Blau, "Wi-Fi Hotspot Networks Sprout Like Mushrooms", *IEEE Spectrum Magazine*, September 2002, Pages: 18-20.

[Bol02] Jeff Boleng, William Navidi and Tracy Camp, "Metrics to Enable Adaptive Protocols for Mobile Ad Hoc Networks", *Proceedings of the International Conference on Wireless Networks (ICWN'02)*, June 24 - 27, 2002, Pages: 293-298.

[Bol02b] Jeff Boleng, "Efficient Network Layer Addressing for Mobile Ad Hoc Networks", In the *Proceedings of the International Conference on Wireless Networks (ICWN '02)*, 2002. Pages: 271-277.

[Bop01] Rajendra V. Boppana and Satyadeva P. Konduru, "An Adaptive Distance Vector Routing Algorithm for Mobile, Ad Hoc Networks", INFOCOM 2001. *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*. Proceedings. IEEE, Volume: 3 , 22-26 April 2001, Pages: 1753 -1762.

[Bra00] Robert Braden, David Clark, Scott Shenker and John Wroclawski, "Developing a Next-Generation Internet Architecture", *Introductory White Paper, The NewArch Project, International Computer Science Institute*, July 2000. http://www.isi.edu/newarch

[Bre00] Lee Breslau, Deborah Estrin, Kevin Fall, Sally Floyd, John Heidermann, Ahmed Helmy, Polly Huang, Steven McCanne, Kannan Varadhan, Ya Xu and Haobo Yu, "Advance in Network Simulation", *IEEE Computer Magazine*, Volume: 33 Issue: 5, May 2000, Pages: 59 -67.

[Bri02] Linda Briesemeister and Günter Hommel, "Localized Group Membership Service for Ad Hoc Networks", *2002 International Conference on Parallel Processing Workshops (ICPPW'02)*, August 18 - 21, 2002, Vancouver, B.C., Canada.

[Bro98] Josh Broch, David A. Maltz, David B. Johnson, Yih-Chun Hu and Jorjeta Jetcheva, "A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols", In *Proceedings of the Fourth Annual IEEE/ACM International Conference on Mobile Computing and Networking (MobiCom'98)*, ACM, Dallas, TX, October 1998. Pages: 85-97.

[But00] Levente Buttyán and jean-Pierre Hubaux, "Enforcing Service Availability in Mobile Ad-Hoc WANs", *First Annual Workshop on Mobile and Ad Hoc Networking and Computing*, 2000. MobiHOC. 2000, First 11 Aug. 2000, Pages: 87 -96.

[But03] Levente Buttyán and Jean-Pierre Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks", *ACM/Kluwer Mobile Networks and Applications*, Vol. 8, No. 5, October 2003.

[Cam02] Tracy Camp, Jeff Boleng and Vanessa Davies, "A Survey of Mobility Models for Ad Hoc Network Research", *Wireless Communication & Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, Vol. 2, No. 5, Pages: 483-502. 2002.

[Cas90] J. D. Case, M. Fedor, M. L. Schoffstall and C. Davin, "A Simple Network Management Protocol (SNMP)", *IETF RFC 1157*, May 1990. http://www.ietf.org

[Cas96] I. Castineyra, N. Chiappa, and M. Steenstrup, "The Nimrod Routing Architecture", *IETF RFC 1992 (informational draft)*, August 1996. http://www.ietf.org

[Cav02] David Cavin, Yoav Sasson and André Schiper, "On the Accuracy of MANET Simulators", *Proceedings of the Second ACM International Workshop on Principles of Mobile Computing*, 2002, Pages: 38 – 43.

[Cha00] Ajay Chandra, V. Gummalla and John O. Limb, "Wireless Medium Access Control Protocols", *IEEE Communications Surveys and Tutorials*, Second Quarter 2000 Vol. 3, No. 2.

[Cha01] Kartik Chandran, Sudarshan Raghunathan, Subbarayan Venkatesan and Ravi Prakash. "A Feedback-based Scheme for Improving TCP Performance in Ad Hoc Networks", *Proceedings of the 18th International Conference on Distributed Computing Systems*, 26-29 May 1998, Pages : 472 -479.

[Cha96] Tushar Deepak Chandra and Sam Toueg, "Unreliable Failure Detectors for Reliable Distributed Systems", *Journal of the ACM*, Volume: 43, Issue: 2, March 1996, Pages: 225-267.

[Cha96b] Tushar Deepak Chandra, Vassos Hadzilacos, Sam Toueg and Bernadette Charron-Bost, "On the Impossibility of Group Membership", In the *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, Philadelphia, Pennsylvania, 1996, Pages: 322 – 330.

[Che04] Yuanzhu P. Chen, Arthur L. Liestman, and Jiangchuan Liu, "Clustering Algorithms for Ad Hoc Wireless Networks", in *Ad Hoc and Sensor Networks*, Edited by Y. Pan and Y. Xiao, Nova Science Publishers, 2004.

[Che98] Tsu-Wei Chen and Mario Gerla, "Global State Routing: A New Scheme for Ad-hoc Wireless Networks", *IEEE International Conference on Communications*, 1998. ICC 98. Conference Record.1998, Volume: 1, 7-11 June 1998, Pages: 171 -175.

[Che99] Wenli Chen, Nitin Jain and Suresh Singh, "ANMP: Ad hoc Network Management Protocol", *IEEE Journal on Selected Areas in Communications*, Volume: 17 Issue: 8, Aug. 1999, Pages: 1506 -1531.

[Cla01] Thomas Heide Clausen, Gitte Hansen, Lars Christensen and Gerd Behrmann, "The Optimized Link State Routing Protocol Evaluation through Experiments and Simulation", The *Fourth International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Aalborg, Denmark, September 9 – 12, 2001.

[Cla02] Thomas Clausen, Philippe Jacquet, Anis Laouiti, Pascale Minet, Paul Muhlethaler, Amir Qayyum, Laurent Viennot, "Optimized Link State Routing Protocol", *IETF MANET Working Group*, December 2002. http://www.ietf.org

[Cla02b] Thomas Heide Clausen, Philippe Jacquet and Laurent Viennot, " Comparative Study of Routing Protocols for Mobile Ad-hoc NETworks", In the *Proceedings of the International Federation for Information Processing Med-Hoc-Net Conference*, 2002.

[Cla02c] Thomas Heide Clausen, Philippe Jacquet and Laurent Viennot, "Comparative Study of CBR and TCP Performance of MANET Routing Protocols", *Workshop on Broadband Wireless Ad-Hoc Networks and Services*, 12-13 September 2002.

[Cof02] K. G. Coffman and A. M. Odlyzko, "Growth of the internet", In *Optical Fiber Telecommunications IV B: Systems and Impairments*, I.P. Kaminow and T. Li. (Editors), Academic Press, 2002, Pages 17-56.

[Cor03] Carlos de Morais Cordeiro, Hrishikesh Gossain and Dharma P. Agrawi, "Multicast over Wireless Mobile Ad Hoc Networks: Present and Future Directions", *IEEE Network*, Volume: 17 Issue: 1, Jan.-Feb. 2003, Pages: 52 -59.

[Cor97] M.S. Corson and V.D. Park, "An Internet MANET Encapsulation Protocol (IMEP) Specification.", *Internet Draft*, http://www.ietf.org, November 1997, (Work in progress).

[Cor99] Scott Corson and Joe Macker, "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations", *IETF RFC 2501*, 1999. http://www.ietf.org

[Cou01] George Coulouris, Jean Dollimore and Tim Kindberg, "Distributed Systems Concepts and Design", *Addison-Wesley*, 2001.

[Das01] Samir R. Das, Charles E. Perkins and Elizabeth M. Royer, "Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks", *IEEE Personal Communications*, Volume: 8 Issue: 1 , Feb. 2001, Pages: 16 -28.

[Das98] S.R. Das, R. Casteneda, Yan Jiangtao and R. Sengupta, "Comparative Performance Evaluation Of Routing Protocols For Mobile, Ad Hoc Networks", *Proceedings of the 7th International Conference on Computer Communications and Networks*, 1998. 12-15 Oct. 1998, Pages: 153 -161.

[Dij59] Edsger W. Dijkstra, "A Note on Two Problems in Connexion with Graphs", *Numerische Mathematik*, Volume 1, 1959, Pages: 269-271.

[Dij74] Edsger W. Dijkstra, "Self-stabilizing Systems in Spite of Distributed Control", *Communications of the ACM*, November 1974, Volume 17, Number 11, Pages: 643-644.

[Dol00] Shlomi Dolev, "Self-stabilization", *MIT Press*, 2000.

[Dov03] Diego Doval and Donal O'Mahony, "Overlay Networks : A Scalable Alternative for P2P", *IEEE Internet Computing*, Vol. 7, No. 3, June/July 2003, pp 2-5.

[Dro97] R. Droms, "Dynamic Host Configuration Protocol", *IETF RFC 2131*, 1997. http://www.ietf.org

[Dub96] Rohit Dube, Cynthia D. Rais, Kuang-Yeh Wang and Satish K. Tripathi, "Signal Stability based Adaptive Routing (SSA) for Ad-Hoc Mobile Networks", *IEEE Personal Communications*, Volume: 4 Issue: 1 , Feb. 1997, Pages: 36 -45.

[Est99] Deborah Estrin, Ramesh Govindan, John Heidemann and Satish Kumar, "Next Century Challenges: Scalable Co-ordination in Sensor Networks", In *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, USA, ACM. August, 1999. Pages. 263-270.

[Fal03] Kevin Fall and Kannan Varadhan (Editors), "The ns Manual (formerly ns Notes and Documentation)", *The VINT Project*, A collaboration between researchers at UC Berkeley, LBL, USC/ISI, and Xerox PARC, Documentation updated daily.

[Fee01] L.M. Feeney, B. Ahlgren, A. Westerlund, "Spontaneous Networking: An Application Oriented Approach to Ad Hoc Networking", *IEEE Communications Magazine* , Volume: 39, Issue: 6, June 2001, Pages:176 – 181.

[Fic03] Faith Fich and Eric Ruppert, "Hundreds of Impossibility Results for Distributed Computing", *Papers in Celebration of the 20th anniversary of PODC*, Distributed Computing, Volume 16, Issue 2-3,September 2003, Pages: 121 – 163.

[Fis85] Michael J. Fischer, Nancy A. Lynch and Michael S. Paterson, "Impossibility of Distributed Consensus with One Faulty Process", *Journal of the Association for Computing Machinery*, Vol. 32, No. 2, April 1985, Pages: 374-382.

[Flo01] Sally Floyd and Vern Paxson, "Difficulties in Simulating the Internet", *IEEE/ACM Transactions on Networking*, Volume: 9 Issue: 4, Aug. 2001, Pages: 392 -403.

[Fly02] Juan Flynn, Hitesh Tewari and Donal O'Mahony, "A Real Time emulation System for Ad Hoc Networks", In *Proceedings of the Communication Networks and Distributed Systems Modelling and Simulation Conference (CNDS 2002)*, San Antonio, Texas, January 27-31, 2002, Pages 115-120.

[For62] Lester Randolph Ford and Delbert Ray Fulkerson, "Flows in Networks", *Princeton University Press*, Oxford University Press, 1962.

[Fre02] Louis E. Frenzel, "Ultrawideband Wireless: Not-So-New Technology Comes Into Its Own", *Electronic Design*, vol. 50, no. 24, Pages: 53-60, Nov., 2002.

[Fro01] Magnus Frodigh, Stefan Parkvall and Christiaan Roobol, "Future-Generation Wireless Networks", *IEEE Personal Communications*, Volume: 8 Issue: 5, Oct. 2001, Pages: 10 -17.

[Gal96] Alessandro Galleni and David Powell, "Consensus and Membership in Synchronous and Asynchronous Distributed Systems", *Laboratory for Analysis and Architecture of Systems* (LAAS), Toulouse, France. Technical Report Number 96104, April 1996.

[Ger00] Mario Gerla, Xiaoyan Hong and Guangyu Pei, "Landmark Routing for Large Ad Hoc Wireless Networks", *IEEE Global Telecommunications Conference*, 2000. GLOBECOM '00, Volume: 3, 27 Nov.-1 Dec. 2000, Pages: 1702 -1706.

[Ger01] Mario Gerla, Rohit Kapoor, Manthos Kazantzidis and Per Johansson, "Ad hoc networking with Bluetooth", In the *Proceedings of the* ACM Workshop on Wireless Mobile Internet (WMI) *at Mobicom*, Rome, Italy, July 2001, Pages: 50 – 55.

[Ger99] Mario Gerla, Ken Tang, Rajive Bagrodia, "TCP Performance in Wireless Multi-hop Networks", *Mobile Computing Systems and Applications*, 1999. Proceedings. WMCSA '99. Second IEEE Workshop on , 25-26 Feb. 1999, Pages: 41 -50.

[Gra96] Stephen Graham and Simon Marvin, "Telecommunications and The City: Electronic Spaces, Urban Places", *Routledge*, London, 1996.

[Gre04] Derek Greene and Donal O'Mahony, "Instant Messaging & Presence Management in Mobile Ad-Hoc Networks", in *Proceedings of the 1st International Workshop on Mobile Peer-to-Peer Computing* (MP2P'04), Orlando, Florida, March 14, 2004

[Gue97] Rachid Guerraoui and André Schiper, "Consensus: The Big Misunderstanding", *6th IEEE Workshop on Future Trends of Distributed Computing Systems* (FTDCS '97), Tunis, Tunisia, October 29 - 31, 1997, Page: 183.

[Gup97] Piyush Gupta and P.R. Kumar, "A System and Traffic Dependent Adaptive Routing Algorithm for Ad Hoc Networks", In the *Proceedings of the 36th IEEE Conference on Decision and Control*, Volume: 3, 10-12 Dec. 1997, Pages: 2375 – 2380.

[Haa97] Zygmunt.J. Haas, "A New Routing Protocol for the Reconfigurable Wireless Networks", Record of the *IEEE 6th International Conference on Universal Personal Communications*, 1997, Volume: 2, 12-16 Oct. 1997, Pages: 562 – 566.

[Hed98] C. Hedrick, "Routing Information Protocol", *IETF RFC 1058*, June 1998. http://www.ietf.org

[Hin98] R. Hinden and S. Deering, "IP Version 6 Addressing Architecture", *RFC 2373*, July 1998.

[Jac01] P. Jacquet, P. Mühlethaler, T. Clausen, A. Laouiti, A Qayyum and L. Viennot, "Optimized Link State Routing Protocol for Ad Hoc Networks", Proceedings of the *IEEE International Multi Topic Conference*, 2001. IEEE INMIC 2001. Technology for the 21st Century, 28-30 Dec. 2001, Pages: 62 -68.

[Jay96] Mahesh Jayaram and George Varghese, "Crash Failures Can Drive Protocols to Arbitrary States", *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing*, Philadelphia, United States, Pages: 247-256.

[Jet01] Jorjeta G. Jetcheva and David B. Johnson, "Adaptive Demand-Driven Multicast Routing in Multi-Hop Wireless Ad Hoc Networks", In the *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Long Beach, CA, USA, 2001, Pages: 33 – 44.

[Joh01] David B. Johnson, David A. Maltz, and Josh Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks", In *Ad Hoc Networking*, edited by Charles E. Perkins, Chapter 5, Pages: 139-172, Addison-Wesley, 2001.

[Joh02] Per Johansson, Rohit Kapoor, Manthos Kazandtzidis and Mario Gerla, "Personal Area Networks: Bluetooth or IEEE 802.11?", *Journal of Wireless Information Networks*, Special Issue On Mobile Ad Hoc Networks (MANETs) Standards, Research, Applications, April 2002.

[Joh02b] David B. Johnson, David A. Maltz, Yih-Chun Hu and Jorjeta G. Jetcheva, "The Dynamic Source Routing Protocol for Mobile Ad Hoc Networks(DSR)", *IETF MANET Working Group*, February 2002. http://www.ietf.org

[Joh96] David B. Johnson and David A. Maltz, "Protocols for Adaptive Wireless and Mobile Networking", *IEEE Personal Communications*, Volume: 3, Issue: 1, February 1996, Pages: 34-42.

[Joh96b] David B. Johnson and David A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks", In *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Chapter 5, Kluwer Academic Publishers, 1996, Pages: 153-181.

[Joh99] Per Johansson, Tony Larsson, Nicklas Hedman, Bartosz Mielczarek and Mikael Degermark, "Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks", In the *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, Seattle, Washington, 1999, Pages: 195 – 206.

[Jon00] Ulf Jonsson, Fredrik Alriksson, Tony Larsson, Per Johansson, Gerald Q. Maguire Jr., "MIPMANET – Mobile IP for MANET", In the *Proceedings of the 1st ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Boston, USA, 2000, Pages: 75-85.

[Jor02] Ramiro Jordan and Chaouki T Abdallah, "Wireless Communications and Networking: An Overview", *IEEE Antennas and Propagation Magazine*, Volume: 44, Issue: 1, Feb. 2002, Pages: 185 – 193.

[Jub87] John Jubin and Janet D. Tornow, "The DARPA Packet Radio Network Protocols", In the *Proceedings of the IEEE*, Volume: 75, Issue: 1, January 1987, Pages: 21-32.

[Kah02] Kevin C. Kahn and David E. Culler, "Ad Hoc Sensor Networks: Proactive Computing Is Coming", *Intel Developer Update Magazine*, Intel Corporation, April 2002.

[Kah78] Robert E. Kahn, Steven A. Gronemeyer, Jerry Burchfiel and Ronald C. Kunzelman, "Advances in Packet Radio Technology", *Proceedings of the IEEE*, vol 66, no 11, November 1978.

[Kan01] Latha Kant, Chavalit Srisathapornphat, Chairporn Jaikaeo and Chun-Hui Zhu, "A Scalable Manage-Agency-based Network Management Design to provide Adaptive Configuration Management for Tactical Battlefield Networks", *Military Communications Conference, 2001. MILCOM 2001. IEEE Communications for Network-Centric Operations: Creating the Information Force,* Volume: 2, 28-31 Oct. 2001, Pages: 928 -932.

[Kap96] Elliott D. Kaplan, "Understanding GPS: Principles and Applications", *Artech House Publishers*, February 1, 1996.

[Kar96] Phil Karn, "MACA- A New Channel Access Method for packet Radio", In the *Proceedings of the ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, Pages: 134-40.

[Kim01] Minkyong and Brian Noble, "Mobile Network Estimation", In the *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, Rome, Italy, Pages: 298 – 309.

[Kwa03] Byung-Jae Kwak, Nah-Oak Song, Leonard E. Miller, "A Mobility Measure for Mobile Ad-Hoc Networks", *IEEE Communications Letters*, Volume: 7, Issue: 8, Aug. 2003, Pages: 379 – 381.

[Lam78] Leslie Lamport, "Time, Clocks, and the Ordering or Events in a Distributed System", *Communications of the ACM*, Volume 21, Number 7, July 1978, Pages: 558-565.

[Lam82] Leslie Lamport, Robert Shostak and Marshall Pease, "The Byzantine Generals Problem", *ACM Transactions on Programming Languages and Systems*, Volume 4, Number 3, July 1982, Pages: 382-401.

[Lan01] Jim Lansford, Adrian Stephens and Ron Nevo, "Wi-Fi (802.11b) and Bluetooth: Enabling Coexistence", *IEEE Network*, Volume: 15, Issue: 5, Sep./Oct. 2001, Pages: 20-27.

[Lao02] Anis Laouti, Paul Mülethaler, Abdellah Najid and Epiphane Plakoo, "Simulation Results of the OLSR Routing Protocol for Wireless Network", *The First Annual Mediterranean Ad Hoc Networking Workshop*, Med-hoc-Net, Sardegna, Italy, Sept 4-6, 2002.

[Lee01] Sung-Ju Lee and Mario Gerla, "Dynamic Load-Aware Routing in Ad hoc Networks", *IEEE International Conference on Communications*, 2001. ICC 2001, Volume: 10 , 11-14 June 2001, Pages: 3206 -3210.

[Lee03] Sung-Ju Lee, Julian Hsu, Russell Hayashida, Mario Gerla and Rajive Bagrodia, "Selecting a Routing Strategy for Your Ad Hoc Network", *Computer Comunications*, Vol. 26, Issue 7, Pages: 723-733, May 2003, Elsevier SciencePublications.

[Lee03b] Seoung-Bum Lee and Andrew T. Campbell, "HMP: Hotspot Migration Protocol for Mobile Ad hoc Networks", *Ad hoc Networks Journal*, Elsevier Publications, Volume: 1, Issue: 1, March 2003.

[Lee99] Sung-Ju Lee, Mario Gerla and Chai-Keong Toh, "A Simulation Study of Table-Driven and On-Demand Routing Protocols for Mobile Ad Hoc Networks", *IEEE Network*, Volume: 13 Issue: 4, July-Aug. 1999, Pages: 48 -54.

[Li00] Jinyang Li, John Jannotti, Douglas S. J. De Couto, David R. Karger, Robert Morris, "Scalable Location Service for Geographic Ad Hoc Routing", In the *Proceedings of the* 6th ACM International Conference on Mobile Computing and Networking (MobiCom '00), Boston, Massachusetts, August 2000, pages 120-130.

[Li01] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee and Robert Morris, "Capacity of Ad Hoc Wireless Networks", In the *Proceedings of the* 7th ACM International Conference on Mobile Computing and Networking, Rome, Italy, July 2001, Pages: 61-69.

[Lun02] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom, and C. Tschudin, "A large-scale testbed for reproducible ad hoc protocol evaluations", *IEEE Wireless Communications and Networking Conference*, 2002. WCNC2002, Volume: 1 , 17-21 March 2002, Pages:412-418.

[Lyn93] Daniel C. Lynch, "Historical Evolution", In *Internet System Handbook*, Daniel C. Lynch and Marshall T. Rose (Editors), Addison-Wesley Company, Reading, Massachusetts, 1993.

[Mac99] Joseph P. Macker and M. Scott Corson , "Mobile Ad Hoc Networking and the IETF", *ACM Mobile Computing and Communications Review*, Vol. 3, Number 2, April 1999.

[Mal01] David A. Maltz, Josh Broch and David B. Johnson, "Lessons from a full-scale multihop wireless ad hoc network testbed", *IEEE Personal Communications*, Volume: 8 Issue: 1, Feb. 2001, Pages: 8 -15.

[Mal94] Dalia Malki, Ken Birman, Aleta Ricciardi and André Schiper, "Uniform Actions in Asynchronous Distributed Systems", *Association of Computing Machinery Annual Symposium on Principles of Distributed Computing*, August 1994, Pages. 274-284.

[Mal99] David A. Maltz, Josh Broch and David B. Johnson, "Experiences Designing and Building a Multi-Hop Wireless Ad Hoc Network Testbed", *CMU School of Computer Science Technical Report CMU-CS-99-116*. March 1999.

[Mal99b] David A. Maltz, Josh Broch, Jorjeta Jetcheva and David B. Johnson, "The Effects of On-Demand Behavior in Routing Protocols for Multi-Hop Wireless Ad Hoc Networks", *IEEE Journal on Selected Areas in Communications*, Volume: 17 Issue: 8 , Aug. 1999, Pages: 1439 -1453.

[Mau01] Martin Mauve, Jörg Widmer and Hannes Hartenstein, "A Survey on Position-based Routing in Ad Hoc Networks", *IEEE Network*, Volume: 15, Issue: 6, Nov/Dec 2001, Pages: 30-39.

[Mca00] A.J. McAuley and K. Manousakis, "Self-Configuring Networks", *MILCOM 2000. 21st Century Military Communications Conference Proceedings*, Volume: 1, 22-25 Oct. 2000, Pages: 315 – 319.

[Mil91] D.L. Mills, "Internet time synchronization: the network time protocol", *IEEE Transactions on Communications*, Volume: 39, Issue: 10, Oct. 1991, Pages:1482 – 1493.

[Mos99] Mitchell L. Moss and Anthony M. Townsend, "How Telecommunications Systems are Transforming Urban Spaces", *Research Report, Taub Urban Research Center*, New York University, April 1999.

[Moy98] J. Moy, "OSPF Version 2", *IETF RFC 2328* (draft standard), 1998. http://www.ietf.org

[Nar02] Luca De Nardis, Pierre Baldi and Maria-Gabriella Di Benedetto, "UWB Ad-Hoc Networks", *IEEE Conference on Ultra Wideband Systems and Technologies*, 2002. Digest of Papers. 2002, 21-23 May 2002, Pages: 219 -223.

[Nes02] Sanket Nesargi and Ravi Prakash, "MANETconf: Configuration of Hosts in a Mobile Ad Hoc Network", *INFOCOM 2002. Proceedings of the Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies.*, Volume: 2 , 23-27 June 2002, Pages: 1059 -1068.

[Nil04] Anders Nilsson, "Performance Analysis of Traffic Load and Node Density in Ad Hoc Networks", *The Fifth European Wireless Conference*, Barcelona, Spain, February 24-27, 2004.

[O'Ma01] D. O'Mahony and L.E. Doyle, "An Adaptable Node Architecture for Future Wireless Networks", *In Mobile Computing: Implementing Pervasive Information and Communication Technologies*, Kluwer Series in Interfaces in OR/CS, Kluwer Publishing, August 2001.

[O'Ma02] Donal O'Mahony and Linda Doyle, "An Adaptable Node Architecture for Future Wireless Networks", in *Mobile Computing: Implementing Pervasive Information and Communication Technologies*, Kluwer series in Interfaces in OR/CS, Kluwer Academic Publishers, 2002, pp77-92.

[O'Ma02b] Donal O'Mahony and Linda Doyle, "Beyond 3G:Fourth-Generation IP-based Mobile Networks, in Wireless IP and Building the Mobile Internet", Editor: Dixit, S., *Artech House*, Norwood, MA, November 2002, ISBN 1-58053-354-X. Chapter 6, pp 71-86.

[Ora90] David Oran, "OSI IS-IS Intra-domain Routing Protocol", *IETF RFC 1142* (draft standard), 1990. http://www.ietf.org

[Par04] Il-kyun bPark, Young-han Kim, and Sang-san Lee, "IPv6 Address Allocation in Hybrid Mobile Ad-Hoc Networks", *Second IEEE Workshop on Software Technologies for Future Embedded and Ubiquitous Systems (WSTFEUS'04)* , Vienna, Austria, May 11 – 12, 2004, Pages: 58 – 62.

[Par97] Vincent D. Park and M.Scott Corson, "A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks", *INFOCOM '97. Proceedings of the Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume: 3, 7-11 April 1997, Pages: 1405 -1413.

[Pei00] Guangyu Pei, Mario Gerla and Tsu-Wei Chen, "FishEye State Routing: A Routing Scheme for Ad Hoc Wireless Networks", *IEEE International Conference on Communications*, 2000. ICC 2000. 2000 Volume: 1, 18-22 June 2000, Pages: 70 -74.

[Per00] Charles E. Perkins, Elizabeth M. Royer and Samir R. Das, "IP Address Autoconfiguration for Ad Hoc Networks", *IETF MANET Working Group*, July 2000. http://www.ietf.org

[Per01] Charles E. Perkins (Editor), "Ad Hoc Networking", *Addison-Wesley Press*, 2001.

[Per02] Dmitri D. Perkins, Herman D. Hughes and Charles B. Owen, "Factors Affecting the Performance of Ad Hoc Networks", *Proceeding of the IEEE International Conference on Communications, 2002. ICC 2002*, Volume 4, Pages: 2048-2052.

[Per02b] Charles E. Perkins, Elizabeth M. Belding-Royer and Samir R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", *IETF MANET Working Group*, November 2002. http://www.ietf.org

[Per94] Charles E. Perkins and Pravin Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers", In *Proceedings of the ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, London, UK, August 1994, Pages: 234-244.

[Per96] Charles E. Perkins, "Mobile-IP, Ad-Hoc Networking, and Nomadicity", In *Proceedings of the 20th International Computer Software and Applications Conference*, 21-23 Aug. 1996, Pages: 472 – 476.

[Per96a] Charles E. Perkins (Editor), "IP Mobility Support" – *IETF RFC 2002*, IBM, 1996. http://www.ietf.org.

[Per99] Charles E. Perkins and Elizabeth M. Royer, "Ad-hoc On-Demand Distance Vector Routing", In *Proceeedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, 1999. WMCSA '99*, 25-26 Feb. 1999, Pages: 90 -100.

[Pic99] Gian Peitro Picco, Amy L. Murphy and Gruia-Catalin Roman, "LIME: Linda Meets Mobility", In the Proceedings of the 1999 International Conference on Software Engineering, 16-22 May 1999, Pages: 368 – 377.

[Poe81] Jon Postel (Editor), "Transmission Control Protocol (TCP)", *IETF RFC 793*, September 1981. http://www.ietf.org

[Qay02] A. Qayyum, L. Viennot and A.Laouiti, "Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks", In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 7-10 January 7-10, 2002, Pages: 3866 – 3875.

[Rah93] Moe Rahnema, "Overview of the GSM System and Protocol Architecture", *IEEE Communications Magazine*, Volume: 31 Issue: 4, April 1993, Pages: 92 -100.

[Raj00] Jyoti Raju and J.J. Garcia-Luna-Aceves, "A Comparison of On-Demand and Table Driven Routing for Ad-Hoc Wireless Networks", *IEEE International Conference on Communications*, 2000. ICC 2000, Volume: 3 , 18-22 June 2000, Pages: 1702 -1706.

[Raj01] Jyoti Raju and J.J. Garcia-Luna-Aceves, "Scenario-based comparison of Source-Tracing and Dynamic Source Routing Protocols for Ad-Hoc Networks", *IEEE International Conference on Communications*, 2001. ICC 2001., Volume: 6 , 11-14 June 2001,Pages: 1919 -1924.

[Ram03] Venugopalan Ramasubramanian, Zygmunt J. Haas and Emin Gün Sirer, "SHARP: A Hybrid Adaptive Routing Protocol for Mobile Ad Hoc Networks", In the *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, Maryland, 2003, Pages: 303 – 314.

[Ric81] G. Ricart and A.K. Agrawala, "An Optimal Algorithm for Mutual Exclusion in Computer Networks", *Communications of theACM*, Volume 24, Number 1, January 1981, Pages:9-17.

[Rig94] Steven E. Rigdon, Emma N. Cruthis and Charles W. Champ, "Design Strategies for Individuals and Moving Range Control Charts", *Journal of Quality Technology*, Volume: 26, Issue: 4, October 1994, Pages: 274-87.

[Rog96] Everett M. Rogers, "Diffusion of Innovations", 4rd Edition, *Free Press*, London.1996.

[Rom01] Gruia-Catalin Roman, Qingfeng Huang and Ali Hazemi, "Consistent Group Membership in Ad Hoc Networks", In the *Proceedings of the 23rd International Conference on Software Engineering*, 12-19 May 2001, Pages: 381 – 388.

[Roy00] Elizabeth M. Royer and Charles E. Perkins, "An Implementation Study of the AODV Routing Protrocol", *IEEE Wireless Communications and Networking Conference, 2000*, Volume: 3, 23-28 Sept. 2000, Pages: 1003 -1008.

[Roy00b] Elizabeth M. Royer, Sung-Ju Lee and Charles E. Perkins, "The Effects of MAC Protocols on Ad Hoc Network Communication", *IEEE Wireless Communications and Networking Conference, 2000*, Volume: 2, 23-28 Sept. 2000, Pages: 543 -548.

[Roy01] Soumya Roy and J.J. Garcia-Luna Aceves, "Using Minimal Source Trees for ON-Demand Routing in Ad Hoc Networks", INFOCOM 2001. *Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Volume: 2, 22-26 April 2001, Pages: 1172 -1181.

[Roy01b] E.M. Royer, P.M. Melliar-Smith and L.E. Moser," An Analysis of the Optimum Node Density for Ad Hoc Mobile Networks", *IEEE International Conference on Communications*, Volume: 3, 11-14 June 2001, Pages:857 – 861.

[Roy02] Elizabeth M. Belding-Royer, "Report on the AODV Interop", University of California at Santa Barbara, U.S.A., *Technical Report 2002-18*, June 2002.

[Roy99] Elizabeth M. Royer and Chai-Keong Toh, "A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks", *IEEE Personal Communications*, Volume: 6 Issue: 2, April 1999, Pages: 46 -55.

[Roy99b] Elizabeth Royer and Charles E. Perkins, "Multicast Operation of the Ad-hoc On-Demand Distance Vector Protocol", *Proceedings of the 5th annual ACM/IEEE international Conference on Mobile Computing and Networking*, 1999, MobiCOM '99, Pages: 207 – 218.

[Sch93] Marco Schneider, "Self-Stabilization", *ACM Computing Surveys*, Volume 25, Issue 1, March 1993, Pages: 45 – 67.

[She02] Chien-Chung Shen, Chavalit Srisathapornphat and Chairporn Jaikaeo, "Adaptive autonomous management of ad hoc networks", *IEEE/IFIP Network Operations and Management Symposium, 2002. NOMS 2002*, 15-19 April 2002, Pages: 891 -893.

[She03] Chien-Chung Shen, Chavalit Srisathapornphat and Chaiporn Jaikaeo, "An Adaptive Management Architecture for Ad Hoc Networks", *IEEE Communications Magazine*, Volume: 41 Issue: 2, Feb. 2003, Pages: 108 -115.

[Shi04] Christopher Shiflet, Elizabeth M. Belding-Royer and Charles E. Perkins, "Address Aggregation in Mobile Ad hoc Networks", *Proceedings of the IEEE International Conference on Communications (ICC)*, Paris, France, June 2004.

[Sil01] Silicon Wave, Inc., "Bluetooth and 802.11 Compared", White Paper, *Published by Silicon Wave, Inc.*, November, 2001. http://www.siliconwave.com

[Siv99] Raghupathy Sivakumar, Prasun Sinha and Vaduvar Brahghavan, "CEDAR: a Core-Extraction Distributed Ad hoc Routing algorithm", *IEEE Journal on Selected Areas in Communications,* Volume: 17, Issue: 8, Aug. 1999, Pages:1454 – 1465.

[Spo01] Marcelo Spohn and J.J. Garcia-Luna-Aceves, "Neighbourhood Aware Source Routing", In the *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, October 4-5, 2001, Long Beach, California, Pages: 11-21.

[Str03] Steve Stroh, "Ultra-wideband: Multimedia Unplugged", *IEEE Spectrum*, Volume: 40, Issue: 9, Sep. 2003, Pages: 23 – 27.

[Su01] William Su, Sung-Ju Lee and Mario Gerla, "Mobility prediction and routing in ad hoc wireless networks", *International Journal of Network Management*, Volume: 11, Number: 1, John Wiley & Sons, January/February 2001, Pages: 3-30.

[Sub99] Madhavi W. Subbarao, "Ad Hoc Networking Critical Features and Performance Metrics", Advanced Wireless Technologies Division, United States' *National Institute of Standards and Technology (NIST),* October 7, 1999.

[Sug00] Masashi Sugano, Tsutomu Araki, Masayuki Murata, Takaaki Hatauchi and Yohei Hosooka, "Performance Evaluation of a Wireless Ad Hoc Network: Flexible Radio Network (FRN)", *IEEE International Conference on Personal Wireless Communications*, 2000, 17-20 Dec. 2000, Pages: 350 -354.

[Sun02] Yuan Sun, Elizabeth M. Belding-Royer and Charles E. Perkins, "Internet Connectivity for Ad hoc Mobile Networks", Journal of Wireless Information Networks Special Issue on "Mobile Ad Hoc Networks (MANETs): Standards, Research, Applications", Volume: 9, Issue: 2, April 2002.

[Sun03] Karthikeyan Sundaresan, Vaidyanathan Anantharaman, Hung-Yun Hsieh and Raghupathy Sivakumar, "ATP: A Reliable Transport Protocol for Ad-Hoc Networks", In *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Annapolis, Maryland, USA, 2003, Pages: 64-75.

[Sun03b] Yuan Sun and Elizabeth M. Belding-Royer, "Dynamic Address Configuration in Mobile Ad hoc Networks", University of California at Santa Barbara, U.S.A., *Technical Report 2003-11*, June 2003.

[Sun04] Yuan Sun and Elizabeth M. Belding-Royer. "A Study of Dynamic Addressing Techniques in Mobile Ad hoc Networks." *Wireless Communications and Mobile Computing*, Pages: 315-329, April 2004.

[Tan96] Andrew S. Tanenbaum, "*Computer Networks*", 1996, Prentice Hall

[Tan99] Ken Tang, Mario Correa and Mario Gerla, "Isolation of Wireless Ad hoc Medium Access Mechanisms under TCP", *Proceedings of the Eight International Conference on Computer Communications and Networks*, 1999. 11-13 October 1999, Pages: 77 -82.

[Tan99b] Ken Tang, Mario Gerla, "Fair Sharing of MAC under TCP in Wireless Ad Hoc Networks", In *Proccedings of Multiaccess, Mobility and Teletraffic for Wireless Communications 1999 Workshop (MMT'99),* October 6th-8th, Venice, Italy, 1999.

[Toh96] Chai-Keong Toh, "A Novel Distributed Routing Protocol To Support Ad-Hoc Mobile Computing", *Conference Proceedings of the 1996 IEEE Fifteenth Annual International Phoenix Conference on Computers and Communications*, 27-29 March 1996, Pages: 480 – 486.

[Ton03] Stephen Toner and Donal O'Mahony, "Self-Organising Node Address Management in Ad-Hoc Networks", in *Springer Verlag Lecture notes in Computer Science 2775*, Springer Verlag, Berlin, 2003, Pages: 476-483.

[Tow00] Anthony M. Townsend, "Life in the Real-Time City: Mobile Telephones and Urban Metabolism", *Journal of Urban Technology*, Volume: 7, Issue: 2, August 2000, Pages: 85-104.

[Tow01] Anthony M. Townsend, "Mobile Computing and Communications: New Interactions Between Information Architecture and Infrastructure Use", Presentation to the *Bringing Information Technology to Infrastructure* Conference, Institute for Civil Infrastructure Systems, Washington D.C., June 2001.

[Tse01] Yu-Chee Tseng, Shih-Lin Wu, Wen-Hwa Liao and Chih-Min Chao, "Location Awareness in Ad Hoc Wireless Mobile Networks", *IEEE Computer*, Volume: 34, Issue: 6, June 2001, Pages: 46 – 52.

[Tur92] John Turek and Dennis Shasha, "The Many Faces of Consensus in Distributed Systems", *IEEE Computer*, Volume: 25, Issue: 6, June 1992, Pages: 8 – 17.

[Uni81] United States' Department of Defence Advanced Research Projects Agency, "Internet Protocol", *RFC 791*, September 1981.

[Vai02] Nitin H. Vaidya, "Weak Duplicate Address Detection in Mobile Ad Hoc Networks", In the *Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking & Computing*, Lausanne, Switzerland, 2002, Pages: 206-216.

[Var92] George Varghese, "Self-stabilization by local checking and correction", Ph.D. Thesis MIT/LCS/TR-583, *Massachusetts Institute of Technology*, 1992.

[Vet94] Ronald J. Vetter, Chris Spell and Charles Ward, "Mosaic and the World-Wide Web", *IEEE Computer*, Volume: 27, Issue: 10, Oct. 1994, Pages:49 – 57.

[Vie03] Laurent Viennot, Philippe Jacquet and Thomas Heide Clausen, "Analyzing Control Traffic Overhead versus Mobility and Data Traffic Activity in Mobile Ad-hoc Network Protocols", *Wireless Networks*, July 2004, Volume: 10, Issue: 4, Pages: 447-455, Kluwer Academic Publishers.

[Wen04] Kilian Weniger and Martina Zitterbart, "Address Autoconfiguration in Mobile Ad Hoc Networks: Current Approaches and Future Directions", *IEEE Network Magazine*, Volume 18, Number 4, Pages: 6-11, July 2004.

[Win98] Moe Z. Win and Robert A. Scholtz, "Impulse Radio: How it works", *IEEE Communications Letters*, Volume: 2, Issue: 2, Feb. 1998, Pages: 36 – 38.

[Xu01] Xu Shugong and Tarek Saadawi, "Does the IEEE 802.11 MAC Protocol Work Well in Multihop Wireless Ad Hoc Networks?", *IEEE Communications Magazine*, Volume: 39, Issue: 6, June 2001, Pages: 130 – 137.

[Xu02] Kaixin Xu,Xiaoyan Hong, and Mario Gerla, "An Ad Hoc Network with Mobile Backbones", *IEEE International Conference on Communications, 2002*, ICC 2002, Volume 5, 28 April-2 May 2002 Pages:3138 – 3143.

[Xu03] Xu Bangnan, S. Hischke and B. Walke, "The Role of Ad Hoc Networking in Future Wireless Communications", In the *Proceedings of the International Conference on Communication Technology*, ICCT 2003, Volume: 2, 9-11 April 2003, Pages:1353 – 1358.

[Yan03] Luqing Yang, Winston K.G. Seah and Qinghe Yin, "Improving Fairness among TCP Flows crossing Wireless Ad Hoc and Wired Networks", In the *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, Annapolis, Maryland, USA, 2003, Pages: 57 – 63.

[Zen98] Xiang Zeng, Rajive Bagrodia and Mario Gerla, "GloMoSim: A Library for Parallel Simulation of Large-scale Wireless Networks", *Parallel and Distributed Simulation*, 1998. PADS 98. Proceedings. Twelfth Workshop on , 26-29 May 1998, Pages:154 – 161.

A Core Ad Hoc Routing Protocols

A.1 Introduction

Chapter 3 presented the concept of a flexible multimodal ad hoc networking system. Such a flexible system is composed of many distinct components. The core ad hoc routing components of such a system are presented in this appendix. Three representative ad hoc routing protocols have been chosen for study from among the many proposed generic routing protocols described in Chapter 2. For the purposes of clarity in other chapters, this appendix simply describes the operational details of the three protocols that have been used in this work. An understanding of, and familiarity with, the operation of these protocols is necessary to appreciate the work presented throughout this thesis. As such, this appendix is useful for those who are not familiar with the operation of AODV, DSR and OLSR.

A.2 Ad Hoc On-Demand Distance Vector Routing - AODV

■ Evolution and Design Objectives

The Ad Hoc On-Demand Distance Vector protocol is a reactive protocol that uses a distance-vector based approach to discover and maintain routes in a network. The protocol has been developed primarily by Charles E. Perkins and Elizabeth M. Belding Royer [Per99]. The protocol builds upon the proactive DSDV protocol. DSDV is a proactive distance-vector algorithm based on the classic Bellman-Ford routing mechanism [For62] that was developed by Charles E. Perkins. DSDV introduced sequence numbers to overcome routing loops that could otherwise frequently occur in mobile ad hoc networks. Each destination entry in the DSDV routing cache is marked with a sequence number that has been assigned by the destination node. The use of sequence numbers allows nodes to distinguish between stale routes and fresh routes. In DSDV, the entire routing table of each node is either periodically broadcast throughout the entire network or else, in an alternative implementation, a partial routing table dump may be triggered by link failures. However, while DSDV is an improvement on the distance-vector protocols used in conventional wired networks, such as RIP, it has been found to be slow to converge in mobile wireless ad hoc networks. The traditional requirement of distance-vector protocols to maintain routing information on the entire network introduces too much overhead into a wireless ad hoc networking system.

Following on from the lessons learned in the development of DSDV, AODV was designed as a '*pure on-demand route acquisition system*' in an attempt to minimise the flooding of control packets by creating routes

on a hop-by-hop demand-only basis, using a distributed distance-vector technique. AODV primarily distinguishes itself from other reactive protocols by virtue of the distributed nature of its route (*path*) discovery procedure. The authors state that the primary aims of the protocol are; to broadcast discovery packets only when necessary, to distinguish between local connectivity management and general topology maintenance and to disseminate information about changes in local connectivity changes to those neighbouring nodes that are likely to need the information.

■ **Operational Description**

The AODV protocol is composed of number of distinct components: path discovery, route cache management, and path and local connectivity maintenance. The data forwarding component of the network-layer is not specified in the AODV literature but is critical to the operation of multi-hop communication in an ad hoc network. In this regard AODV follows the design of conventional networks where data forwarding is not the function of the routing protocol. Instead, it is handled in accordance with the specifications of the IETF RFC 1812 [Bak95], which details the requirements of an IPv4 router. Only the basic operations of these components are outlined here; detailed descriptions of the protocol may be found in [Per99], [Per01], [Per02b] and [Roy00].

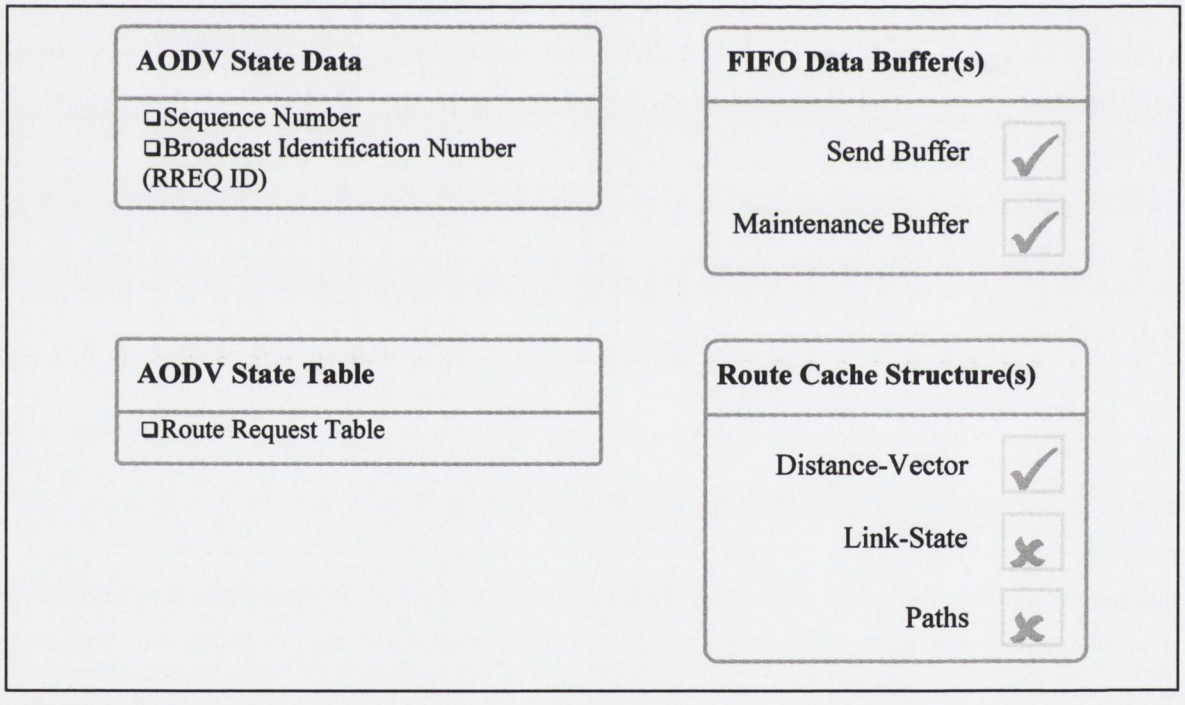


Figure A-1 AODV Protocol State Data, Tables, Route Cache and Data Buffers

Each AODV node maintains protocol state tables, state data, data buffers and a route cache; these are listed in Figure A.1. Each AODV node maintains its own Sequence Number, which appears in the route tables of other nodes as the Destination Sequence Number. Sequence Numbers are used to guarantee loop-freedom of routes towards any destination as they allow nodes to distinguish stale and fresh routing information.

The Broadcast ID Number is used to uniquely distinguish each broadcast packet sent by the node. Each node maintains a Route Request Table that allows it to avoid processing broadcast RREQ packets that it has

already dealt with. Each node also maintains two FIFO data buffers for data packets waiting for routes to be discovered and data packets in transit awaiting an acknowledgement from the next hop node. Each AODV node maintains a distance-vector based cache for each Destination Address entry that contains information based on the most recent sequence number for that Destination, i.e. its Destination Sequence Number. This information also includes a Hop Count, i.e. the distance from the node to the Destination Address. Each of these AODV protocol elements will be described in further detail as the discussion of the protocol progresses.

Path Discovery:

As noted, one of the key features of AODV is that the discovery of routes is done on a reactive basis. When a source node requires a route to a destination, for which no valid route is found in the source node's Route Cache, the *path discovery* process is invoked. The source node broadcasts a route request message, **AODV_Route_Request**, to its neighbours. Before originating the request message, the source node buffers the waiting data packets, which have prompted the path discovery process, in a first-in, first-out (FIFO) Send Buffer. The source node also increases its own unique monotonically increasing sequence number which it then includes in the request message Originator Sequence Number field. The use of a new (increased) sequence number in each request ensures that the other nodes treat the current request as the latest information on the source node.

The details of the **AODV_Route_Request** message are shown in Figure A.2. Its RREQ ID is dictated by the router's own unique monotonically increasing identification number for broadcast messages. The RREQ ID allows other nodes to distinguish between new route requests and old ones by looking at the unique combination of the **AODV_Route_Request**'s Originator Address and RREQ ID. Each node has a unique address, which it uses to fill in the Originator Address of the **AODV_Route_Request**. The node's address is chosen either manually or by another automated scheme independently of the AODV protocol. A record of all **AODV_Route_Requests**, based on the Originator Address and RREQ ID, is kept in the receiving node's Route Request Table for a limited time so that broadcast **AODV_Route_Request** flooding storms may be quenched.

Subject to the limiting effects of the time-to-live (TTL) of the network-layer packet encapsulating the **AODV_Route_Request** packet, each neighbour then rebroadcasts the **AODV_Route_Request** to their neighbours until the destination is found or until an intermediate node with a route to the destination is found. In the example illustrated in Figure A.3a, node 1 seeks a route to node 20. Node 1 originates an **AODV_Route_Request** seeking node 20, by broadcasting the message to its neighbouring nodes, in this case nodes 6 and 9. The request message is rebroadcast until it gets to node 20 as none of the other nodes have fresh routing information.

If the node has a stale entry for this Destination Address in its Route Cache, it will have an associated Destination Sequence Number in the cache also. If so, the node places this most recently known Destination Sequence Number for the destination node into the **AODV_Route_Request**. AODV does not delete routing information from the cache as soon as it times out. Rather, it holds the information in soft-state, i.e. it

is not used for routing packets but appears in the Route Cache, so that it may be used in the route discovery process. If the Destination Sequence Number is unknown, i.e. there is no entry for the Destination Address in the Route Cache, the source node sets the U Bit in the request message. The inclusion of the Destination Sequence Number in the `AODV_Route_Request` enables intermediate nodes to tell if the information that they hold on the destination node is fresher than the information that the source node holds. Also, if the source node only wants the destination node to respond to the `AODV_Route_Request`, it may set the D Bit.

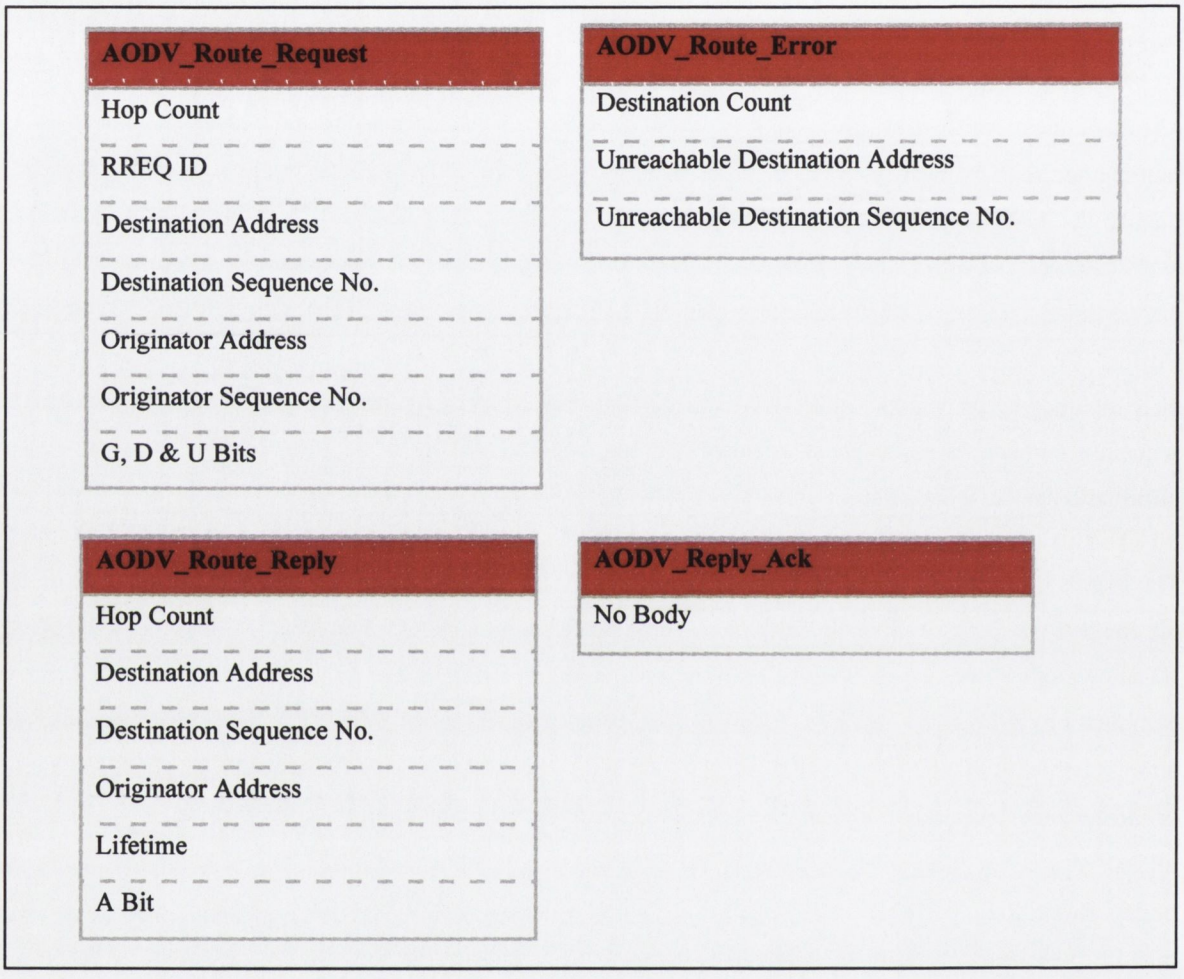


Figure A-2 AODV Control Messages

In order to control and limit the number of broadcast request messages that are originated and forwarded in the network, AODV nodes use an *expanding ring* search technique. This technique prevents unnecessary network-wide dissemination of `AODV_Route_Requests`. The idea of the expanding ring search is that a node should initially originate a request with a very low TTL which is specified in the encapsulating network-layer packet. A low TTL ensures that the request message does not spread too far, decreasing the possibility of it causing congestion. It allows a node to query nearby nodes for routing information. If the request fails, i.e. no reply is received within some time limit, then the source node repeats the request message a limited number of times, incrementally increasing the TTL each time up to a maximum limit, e.g. the permitted network diameter. The rate at which AODV nodes issue request messages is also limited so that a single node cannot overwhelm the network with broadcast messages for multiple destination nodes.

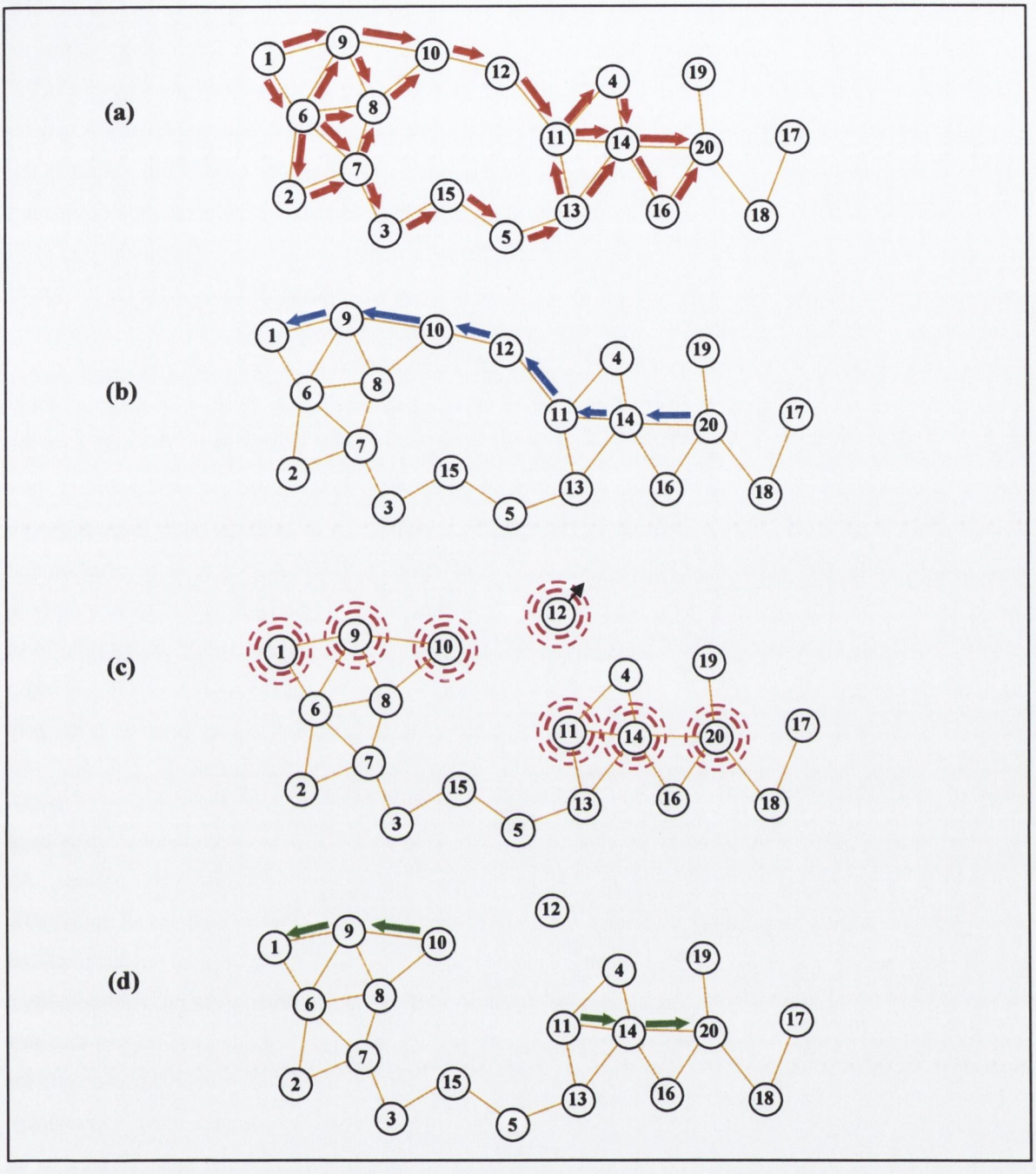


Figure A-3 AODV Networking Example

Nevertheless, as the **AODV_Route_Request** propagates through the network each intermediate node creates a short-lived *reverse path* in its Route Cache if the Source Sequence Number indicates that the information, regarding the source node, is fresh enough. The reverse path is a path back to the source node that may be used by the **AODV_Route_Reply** message: in effect the reverse path is a record of the neighbouring node from which the first copy of each unique **AODV_Route_Request** was received. The reverse path information is only kept in the Route Cache for enough time to allow an **AODV_Route_Request** to traverse the network and possibly generate an **AODV_Route_Reply** back along the path. In the example illustrated in Figure A.3a, node 12 would record node 10 as being the next hop node on the path to the source of the **AODV_Route_Request**, node 1. Intermediate nodes may only reply to the **AODV_Route_Request** if they have a fresh route to the destination in their Route Cache, i.e. one with a Destination Sequence Number equal to, or greater than, the Destination Sequence Number in the

request message. If the Destination Sequence Number in the message is higher than the value recorded by the intermediate node, then the intermediate node deletes its existing route entry for the Destination Address as its information is stale and retransmits the **AODV_Route_Request** subject to the hop limit specified in the encapsulating network-layer packet. If an intermediate node originates the **AODV_Route_Reply** it may be required to send a *gratuitous* **AODV_Route_Reply** to the destination node too if the G Bit was set in the request message. If intermediate nodes were to reply to all requests from a source node, then the destination node may never learn of a path to the source node. Many communication streams are bidirectional, e.g. TCP, and delay may be incurred by the destination node having to invoke a path discovery process for the return route.

When a request reaches either the destination or an intermediate node with fresh information a reply message, **AODV_Route_Reply**, is unicast back to the source along the reverse path. The nodes on the reverse path that receive the **AODV_Route_Reply** message create longer-lived *forward paths* in their Route Caches to the node that originated the **AODV_Route_Reply**. The value of the timer associated with the forward-path is specified by the Lifetime field in the reply message. Figure A.3b depicts the situation that would occur if the first request to reach node 20 had come from node 14. Node 20 would issue a reply to node 14 and ignore the later identical request from node 16. In Figure A.3b, node 11 would record node 14 as being on the forward path to node 20. When an **AODV_Route_Reply** reaches the source node it may begin forwarding data that is pending in the Send Buffer, thus completing the path discovery process. If the path discovery process fails, the node dumps the data packets waiting in the FIFO Send Buffer.

One other feature of the path discovery process concerns the ability of AODV to overcome the existence of asymmetric links, or general link failures, in the network during the path discovery process. An **AODV_Route_Reply** may fail to reach the source node if the path that it traverses contains an asymmetric link. If no other reply has been generated, the source node will have to wait until its original request times out before issuing another request. This cycle may be repeated until the source node gives up. Additionally, a node that has replied to a request over an asymmetric link will ignore subsequent identical request that may arrive over symmetrical links. To ensure that a reply is delivered to the source node, a replying node sets the A Bit in the **AODV_Route_Reply** that it issues. A node that receives this request must acknowledge receipt of the message by sending an **AODV_Reply_Ack** message. If a node fails to receive an acknowledgement it will add the next hop, to which it had sent the **AODV_Route_Reply**, to a blacklist and will subsequently ignore request from that node.

Route Cache Management:

Both the reverse path and forward path route entries have associated timers, the expiration of which will cause the paths to be expunged from the Route Cache. The reverse paths' timers are short-lived and they are expunged if a reply is not unicast back along the path within a short time. As depicted in Figure A.3a, many nodes will have set up reverse paths during the path discovery phase but only a few of those nodes are actually on the discovered path; nodes 1, 9, 10, 12, 11, 14 and 20. The other reverse paths will be deleted as they will not be used by node 1.The longer-lived forward path's timer expires if the route is not used and the timer is not refreshed; a route cache entry's timer is refreshed whenever a node forwards data along its path.

A node will only update its routing information based on an **AODV_Route_Reply** if the Destination Sequence Number in the message is greater than the one recorded in the Route Cache or if the Destination Sequence Numbers are equal but the Hop Count in the message is lower.

Data Forwarding:

The AODV specification documents do not dictate any mechanisms to handle the forwarding of data. The protocols authors consider AODV to be a route cache management protocol, expecting that data forwarding is handled separately in accordance with [Bak95]. In light of that, the forwarding of data in the implemented AODV network is done on a hop-by-hop basis using the entries in the route cache. The route cache used in the AODV protocol is designed to accommodate the distance-vector routing methodology used in the protocol.

Local Connectivity and Path Maintenance:

Nodes on an active route, i.e. a route over which traffic is being forwarded, generate periodic Hello messages. These messages are actually **AODV_Route_Reply** messages that have broadcast with a TTL of 1 in the encapsulating network-layer packet, ensuring that they are not retransmitted beyond the one-hop neighbourhood. Nodes on an active route listen out for Hello messages from their neighbours and if a Hello message is not received within a defined period it is assumed that the link is broken and the node proceeds to delete dependent route entries and either send **AODV_Route_Error** messages to appropriate nodes or attempt to repair the link locally.

For each route maintained by a node's route cache, a node maintains a list of *precursors*. Precursors are nodes that may be using the node to forward packets. The list of precursors for a destination route entry contains nodes to which an **AODV_Route_Reply** was forwarded or generated. **AODV_Route_Error** messages are sent to nodes on the precursor list. In some cases, nodes may choose to locally repair a broken path by issuing a limited fresh **AODV_Route_Request** for the destination node. In Figure A.3c, each node along the active path periodically beacons a Hello message to its neighbours but node 12 moves out of range of its neighbouring nodes 10 and 11. As node 12 has moved away, nodes 10 and 11 will notice this link failure and send **AODV_Route_Error** messages to nodes 9 and 14, the precursor nodes of 10 and 11 respectively, as these nodes were using nodes 10 and 11 to reach node 12. As depicted in Figure A.3d, nodes 9 and 14 will forward the **AODV_Route_Error** messages on to nodes 1 and 20 which are their respective precursor nodes. When the error message reaches nodes 1 and 20, these nodes may chose to initiate a new route discovery process is they still need to use the route that has been reportedly broken.

During the local repair process, which should not be noticed by the source node, a FIFO Maintenance Buffer holds data packets that were en route to their destination. Pending the reestablishment of the route, the packets are held in this buffer for a limited duration. If the local repair process fails, the buffered data packets are dropped.

■ **Additional Features**

Multicasting:

The authors of AODV have, in another specification document [Roy99b], extended the protocol for multicasting use. The Multicast AODV (MAODV) protocol requires some additional protocol state to be maintained at each node to allow nodes to form multicast trees and groups. However, MAODV is still reliant on the same underlying AODV functionality described above.

Heterogeneous Networks:

The AODV specification recognises the fact that an ad hoc network may have connections to other networks, or *routing domains,* that do not use AODV. However, the specification does not elaborate on any mechanisms to enable or maintain connectivity with such other networks.

A.3 Dynamic Source Routing - DSR

■ **Evolution and Design Objectives**

The Dynamic Source Routing protocol is a purely reactive protocol, the operation of which differs from most other approaches to dynamic routing in networks. The protocol, as its name implies, uses a dynamic source-initiated routing technique to get packets from a source to a destination in a mobile wireless ad hoc network. DSR represents a step-change in the design of routing protocols for mobile ad hoc networks and was designed by David Johnson, David Maltz and Josh Broch at Carnegie Mellon University [Joh01]. The authors were influenced by the design of IEEE 802 (i.e. link-layer) source routing bridges.

The DSR specification document [Joh02b] states that the "*protocol is designed for mobile ad hoc networks with up to around two hundred nodes, and is designed to cope with relatively high rates of mobility.*" The authors also assume that the network diameter will often be small, e.g. 5 to 10 hops. It is also assumed that "*the speed with which the nodes move is moderate with respect to the packet transmission latency and wireless transmission range of the particular underlying network hardware in use.*" Also, while not compulsory, the protocol has been designed to take advantage of radio hardware for broadcast media that allow nodes to operate in a promiscuous receive mode. This mode enables the nodes to access all packets, regardless of the link-layer address, allowing the protocol to take advantage of '*sniffed*' data. Only the DSR header and encapsulated network-layer header are sniffed; the data payload is not accessed, allowing its integrity to be secured on an end-to-end basis.

■ **Operational Description**

The DSR protocol is composed of a number of distinct components; route discovery, route maintenance, route cache management and data delivery. Only the basic operations of these components are outlined here; detailed descriptions of the protocol may be found in [Joh02b] and [Joh01]. Unlike most other routing

protocols that follow the IETF's RFC 1812 guidelines for conventional routers when it comes to data forwarding, DSR employs a distinctive source-routing technique.

Each DSR node maintains protocol state tables, state data, data buffers and a route cache; these are listed in Figure A.4. The Broadcast ID Number is used to uniquely distinguish each broadcast packet sent by the node. Each node maintains a Route Request Table that allows it to avoid processing broadcast RREQ packets that it has already dealt with. The BlackList Table and the Gratuitous Route Reply Table are used in the route discovery process. Each of these DSR protocol elements will be described in further detail as the discussion of the protocol progresses.

Route Discovery:

As noted, one of the key features of the DSR is that all aspects of the protocol operate entirely on-demand. A node only invokes the *route discovery* mechanism when it requires a route to a given destination for which it does not have a valid route in its Route Cache. The source node broadcasts a route request message, `DSR_Route_Request`, to its neighbours indicating that it seeks a route to the specified Target Address. Before originating the request message, the source node buffers the waiting data packets which have prompted the path discovery process, in a first-in, first-out (FIFO) Send Buffer. The RREQ ID in the `DSR_Route_Request` message, the details of which are shown in Figure A.5, is dictated by the router's unique monotonically increasing identification number for broadcast messages.

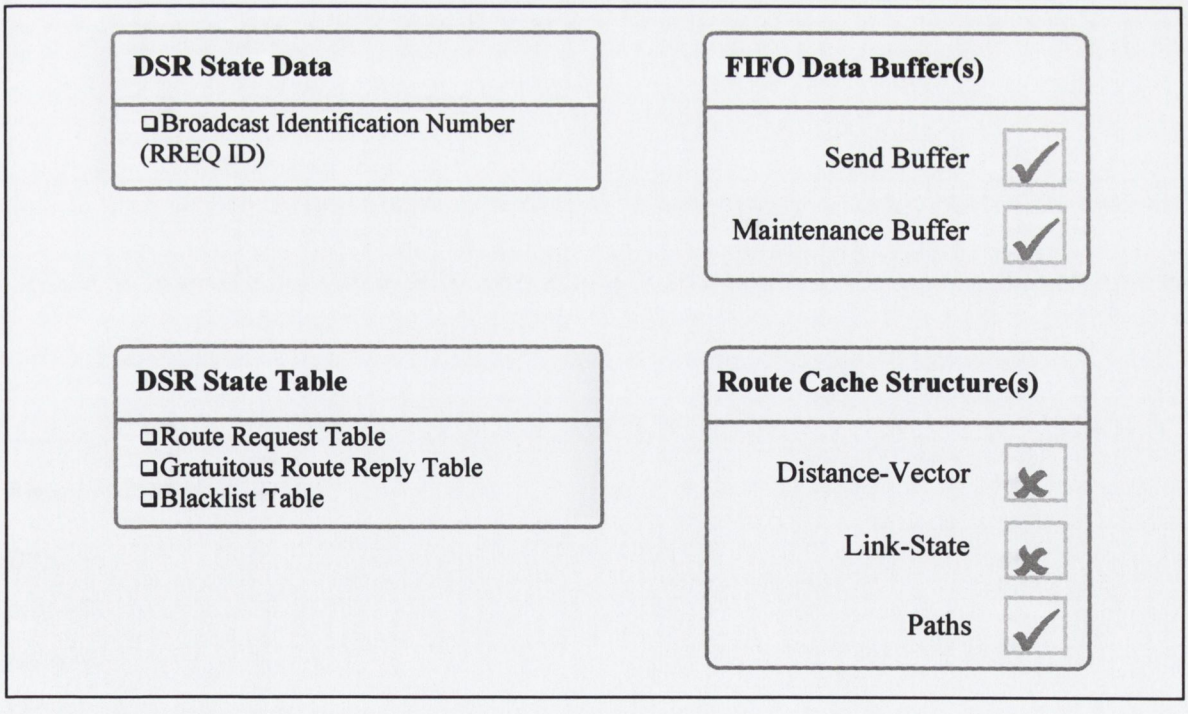


Figure A-4 DSR Protocol State Tables, State Data, Route Cache and Data Buffers

The RREQ ID allows other nodes to distinguish between new route requests and old ones by looking at the unique combination of the `DSR_Route_Request's` Source Address and RREQ ID. The Source Address is not found in a DSR-specific packet, rather it is found in the encapsulating network-layer packet. A record of all `DSR_Route_Requests`, based on the Source Address and RREQ ID, is kept in the Route Request

Table for a limited time so that `DSR_Route_Request` flooding storms may be quenched. Subject to the limiting effects of the TTL of the `DSR_Route_Request` packet, each neighbour then rebroadcasts the `DSR_Route_Request` to their neighbours until the destination is found or until an intermediate node with a route to the destination is found. In the example illustrated in Figure A.6a, node 1 seeks a route to node 20. Node 1 originates a `DSR_Route_Request` seeking node 20 by broadcasting the message to its neighbouring nodes, in this case nodes 6 and 9. The request message is rebroadcast until it gets to node 20 as none of the other nodes have fresh routing information.

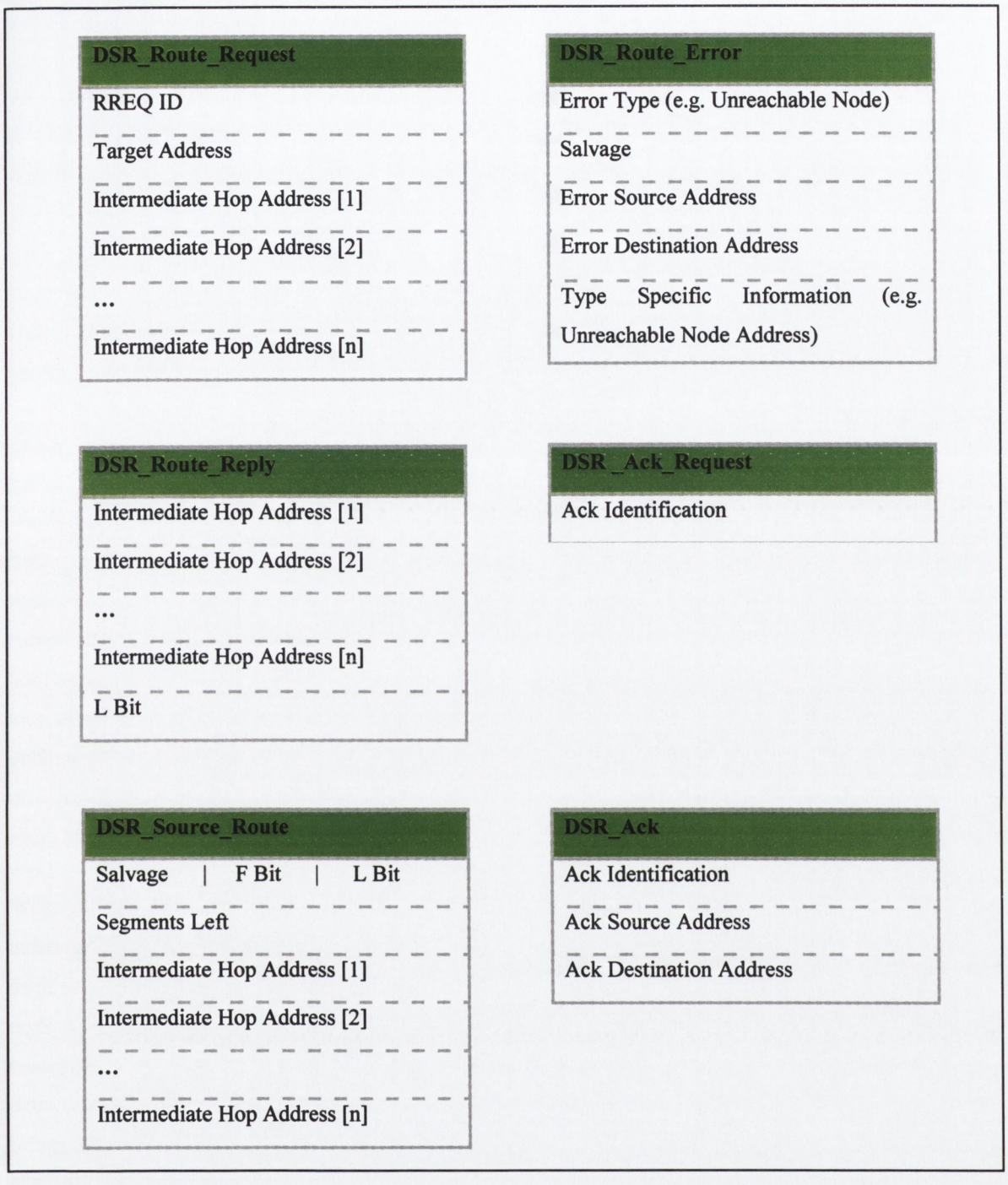


Figure A-5 DSR Control Messages

DSR employs two techniques to control the flooding of `DSR_Route_Requests` through the network: *propagating* requests and *non-propagating* requests. The hop limit of each `DSR_Route_Request`, specified in the TTL field of the encapsulating network-layer packet, limits the number of intermediate nodes that the request may be forwarded to. The hop limit is decremented whenever the request is forwarded. Only requests bearing a non-zero hop limit may be rebroadcast. DSR nodes issue non-propagating requests, i.e. the hop limit is already set to zero, when they only wish to query their neighbours. This is an inexpensive technique that allows a node to see whether the destination node is one of its neighbours or, if not, to see if any of its neighbours know of a valid route to the destination. If the non-propagating request fails, then the source node may issue a propagating request. Propagating requests are issued with a hop limit greater than zero, thus allowing the request to be forwarded to nodes beyond the neighbourhood of the node. Nodes may

also employ an *expanding ring* technique when issuing propagating requests: if a node reissues a request it increases the hop limit subject to an upper limit.

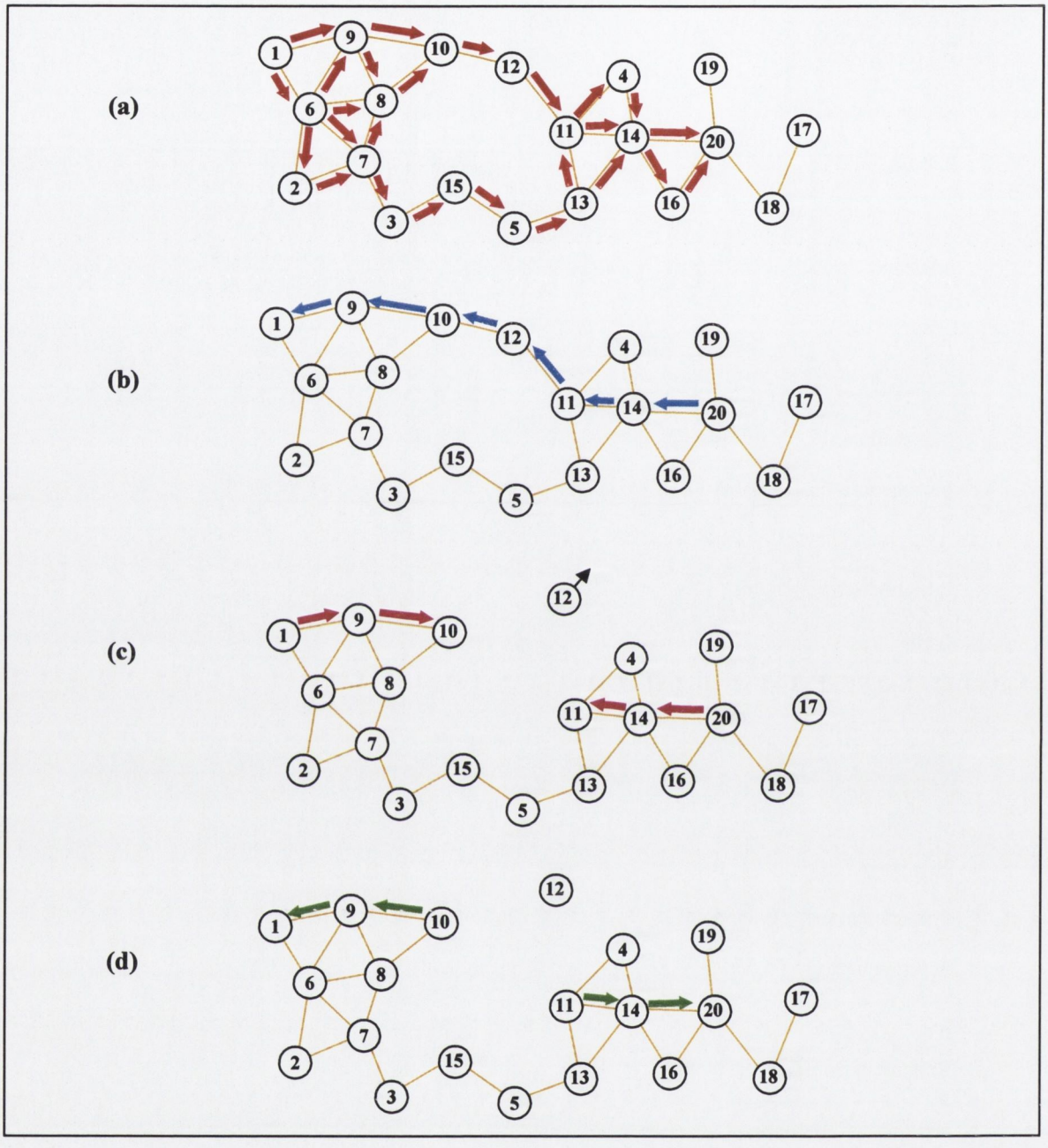


Figure A-6 DSR Networking Example

As the `DSR_Route_Request` propagates through the network each intermediate node, that cannot generate a reply message, records its own address in the request message as the next Intermediate Hop Address, after checking that its own address does not already appear in the route record. This route record would have been initialised as an empty list by the initiator of the request. When the `DSR_Route_Request` gets to the target node, e.g. node 20 in Figure A.6a, the target node returns a `DSR_Route_Reply` to the node that was the initiator of the `DSR_Route_Request`. The `DSR_Route_Reply` contains a copy of the route record that had accumulated in the

DSR_Route_Request. The target node, and each intermediate node, records the route record in their own Route Cache for possible later use.

In some cases an intermediate node may have cached information about a route to the requested Target Address. In such an event, instead of rebroadcasting the **DSR_Route_Request** message, the intermediate node uses its cached routing information to fill out the rest of the route record contained in the **DSR_Route_Request**. It then checks the entire route record to ensure that there is no duplication of addresses, i.e. there are no routing loops. The intermediate node then returns a **DSR_Route_Reply** to the source node. One other feature of the path discovery process concerns the ability of DSR to overcome the existence of asymmetric links, or general link failures, in the network during the path discovery process.

Some underlying link-layer protocols require that links are symmetric in nature for unicast transmission. In such a scenario DSR can test the symmetry of the discovered route by sending the reply back over the reversed route. For example, in the case of the network of Figure A.6b, node 20 would send the **DSR_Route_Reply** back to node 1 over the discovered route through nodes 14, 11, 12, 10 and 9. This route was chosen as first request to reach node 20 came from node 14 in this example. The reply would fail to reach node 1 if any of the links was asymmetric. If a DSR node discovers that a node is asymmetric, when the link-layer requires that links are symmetric, it may list the asymmetric link in a Blacklist Table so that it may avoid using it to forward **DSR_Route_Reply** messages. If the link-layer protocol can operate under both symmetric and asymmetric links then the reply may be returned to the source node by piggybacking it on a new **DSR_Route_Request** for the source node, i.e. the original target node now tries to discover a route to the original source node.

Whenever a node processes a **DSR_Route_Request** or **DSR_Route_Reply** message it checks the message's route record for fresh routing information, regardless of whether the message was intended for the node or not. For instance, if node 13 in Figure A.6b overheard the **DSR_Route_Reply** from node 20 to node 1 as it was forwarded by nodes 14 and 11, it would be able to cache the route record. Such a route record should allow node 13 to contact any node on the route from node 1 to node 20 without engaging any extra control overhead. Whenever the information in the Route Cache is updated, whether it is from information taken from a **DSR_Route_Reply** message or overheard information, then the Send Buffer is checked for data packets that may be awaiting the new routing information. Otherwise, data packets are dropped from the Send Buffer after a defined waiting period.

Route Cache Management:

Each route stored in the cache of a DSR node has an associate timer. The timer is refreshed whenever a route is used to forward data. Routes are deleted from the cache when the timer expires. New route discovery processes are invoked if the route is required again.

Data Forwarding:

Unlike most other protocols, pure DSR does not require intermediate nodes to maintain any routing information. The basis of source-routing, the technique used by DSR, is that the source node is responsible

for gathering routing information that it then places in the `DSR_Source_Route` header of the data packet. As shown in Figure A.5, this header contains the entire record of the route from the source node to the destination node. As the data packet is unicast from node to node, intermediate nodes simply look at the source route record in the header to determine the next hop in the path. The use of source routing means that there is no requirement for intermediate nodes to maintain routing information about destination nodes that they are not currently routing data to.

DSR also engages a *route shortening* mechanism that a node invokes whenever it overhears a packet for which it was not the intended next-hop. If the receiving node notices that its address is mentioned later in the, as yet, unvisited section of the route record, then it may send a gratuitous `DSR_Route_Reply` message to the source of the data packet informing it of the shorter route that now exists. The node also keeps a record of the gratuitous reply in its Gratuitous Route Reply Table so it may limit the number of gratuitous replies it sends to any source node.

Route Maintenance:

Route maintenance is the mechanism that enables a source node to detect, while source-routing a packet, whether the network topology has changed to such an extent that the existing route does not work anymore. Each node that either originates or forwards a source routed packet is responsible for ensuring that the packet is received by the next hop on the specified path. A node buffers the packet in its Maintenance Buffer pending an acknowledgement that the packet was successfully forwarded. Confirmation of receipt by the next hop may come in three forms; link-layer acknowledgment, active network-layer acknowledgement or passive network-layer acknowledgement. Link-layer acknowledgments may be used if cross-layer feedback is possible and the link-layer provides confirmation of unicasting success. If using passive network-layer acknowledgments, which presupposes promiscuous packet reception ability, a node listens for the forwarding of the data packet by the node to which it had forwarded the packet. For example, in Figure A.6c, node 10 would have forwarded the data packet to node 12, expecting to overhear node 12 forwarding the packet to node 11. Failure to overhear such a transmission would indicate that the link had failed. The node may attempt to send the packet a number of times before it is satisfied that the link has failed. Passive acknowledgements cannot be used on the last hop as the destination node will not be forwarding the data packet. The last option is to use active network-layer acknowledgements. In this case, a forwarding node requests an acknowledgement from the next node in the route by inserting a `DSR_Ack_Request` into the packet header. The receiving node is expected to respond with a `DSR_Ack`. Again, a node may resend the source-routed packet and `DSR_Ack_Request` a number of times before inferring that the link has failed.

If a node determines that a link has failed it originates a `DSR_Route_Error` packet to the source node of the packet that failed to be routed, in this case an *Unreachable Node* error message. As depicted in Figure A.6d, nodes 10 and 11 would send error messages to nodes 1 and 20 if either of these nodes was the source of the data packet that failed to be routed across the broken link. Any node that receives the error message removes entries from its Route Cache that depend on the failed link specified in the message.

■ **Additional Feature**

Heterogeneous networks:

The DSR specification documents recognise the possibility that a DSR network may have connections to an external network that does not operate the DSR protocol. While the documents do not detail the manner in which a network should interact with an external network, they do mandate that certain state information be maintained by the nodes to accommodate the presence of external networks. The L (last hop) Bit in the `DSR_Route_Reply` message allows nodes to identify nodes as being gateways between networks. Corresponding flags must also be maintained in the Route Cache to identify such nodes. The F (first hop) and L Bits in the `DSR_Source_Route` header may also allow data to be routed from and to gateway nodes.

A.4 Optimised Link State Routing - OLSR

■ **Evolution and Design Objectives**

The Optimised Link State Routing protocol builds on the prior work that was done in the development of stable link state routing protocols such as OSPF [Moy98]. The protocol was designed by a group of researchers at the INRIA, the national French research body [Cla02]. OLSR optimises the classical link state protocol for the resource-constrained mobile wireless ad hoc networking environment by reducing the size of its control packets and by minimising the flooding of these control packets by selecting an optimal subset of nodes, called *multipoint relays* (MPRs), to retransmit and diffuse these packets. The OLSR protocol is only concerned with the discovery of routing information and does not specify data forwarding mechanisms. This approach is in keeping with the conventional routing protocols.

The proactive nature of the protocol means that nodes have routes available immediately, whether or not they are needed. As the protocol maintains routes to all nodes in the ad hoc network it is said to be more suitable for "*large and dense ad hoc networks*". It is also noted by the authors that the protocol is suited to "*networks, where the traffic is random and sporadic between 'several' nodes rather than being almost exclusively between a small specific set of nodes*". The authors have specifically targeted this protocol towards large, dense networks for which the multipoint relay optimisation is most suited [Qay02].

■ **Operational Description**

The OLSR protocol collates and disseminates network information in three main stages: the sensing of neighbours, the selection and declaration of MPRs and the dissemination of topology control messages. In conventional networks data forwarding is not the function of the routing protocol, it is handled in accordance with the specifications of the IETF RFC 1812, which details the requirements of an IPv4 router. Only the basic operations of these components are outlined here; detailed descriptions of the protocol may be found in [Jac01] and [Cla02].

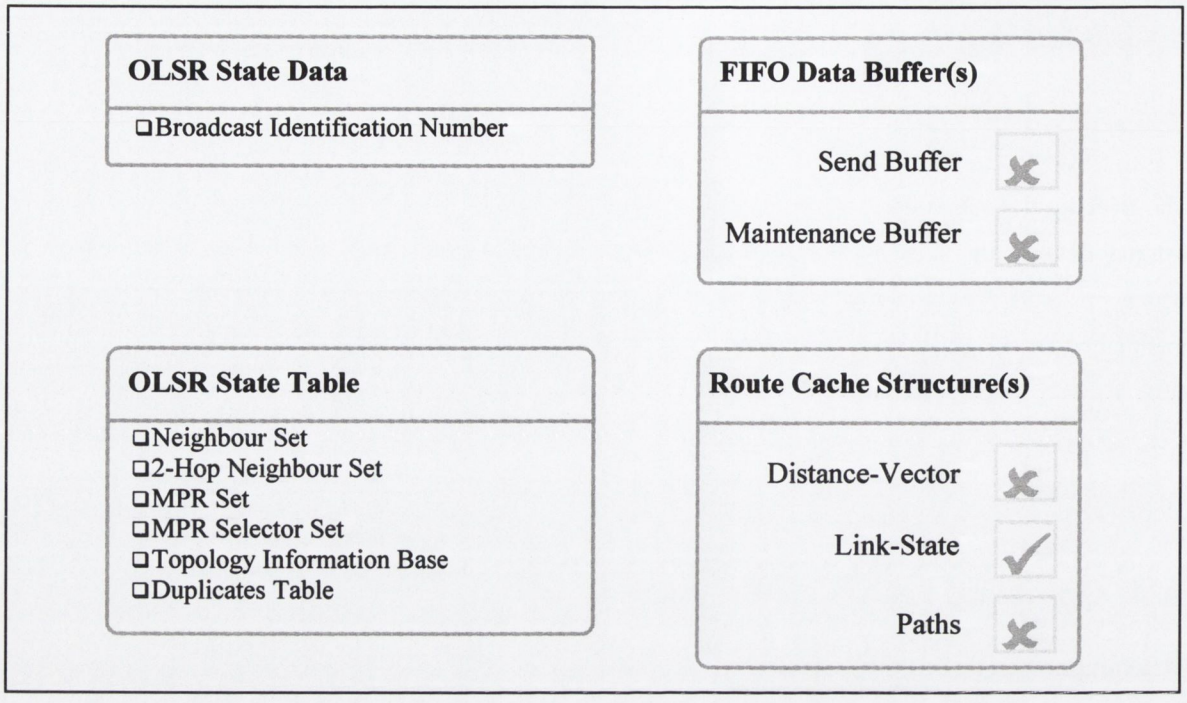


Figure A-7 OLSR Protocol State Tables, State Data, and Route Cache

Each OLSR node maintains protocol state tables, state data and a route cache; these are listed in Figure A.7. OLSR nodes do not maintain any FIFO buffers for data packets waiting on routes or for packets awaiting acknowledgements in transit. All OLSR control packets are broadcast and contain a unique Broadcast Identification Number. OLSR nodes maintain a Duplicates Table that allows them to distinguish control packets that they have already processed. The Neighbour Set, 2-Hop Neighbour Set, MPR Set, MPR Selector Set and Topology Information Base tables are used by the OLSR node to track the neighbourhood nodes, to select relay nodes and to share topology information. The OLSR Route Cache contains link-state entries based on the information in these state tables. Each of these OLSR protocol elements will be described in further detail as the discussion of the protocol progresses.

Neighbour Sensing:

A periodic neighbour sensing mechanism allows an OLSR node to learn of the nodes in its neighbourhood and to tell with which nodes it has a symmetric (bi-directional) or asymmetric (unidirectional) link, i.e. it allows a node to determine the link-states. Establishing that such a link is symmetrical requires that the link is checked in both directions. `OLSR_Hello` messages are used to establish link quality information, see Figure A.8 for the format of the Hello message. The `OLSR_Hello` message contains information about a node's neighbours and their link quality. The `OLSR_Hello` message contains a list of symmetric links, i.e. those neighbouring nodes with which there is bi-directional communication, asymmetric links, i.e. those neighbouring nodes that have been heard by this node but that have not declared that they have heard from this node, and MPR nodes, i.e. neighbouring nodes that this node has chosen to be in its MPR Set.

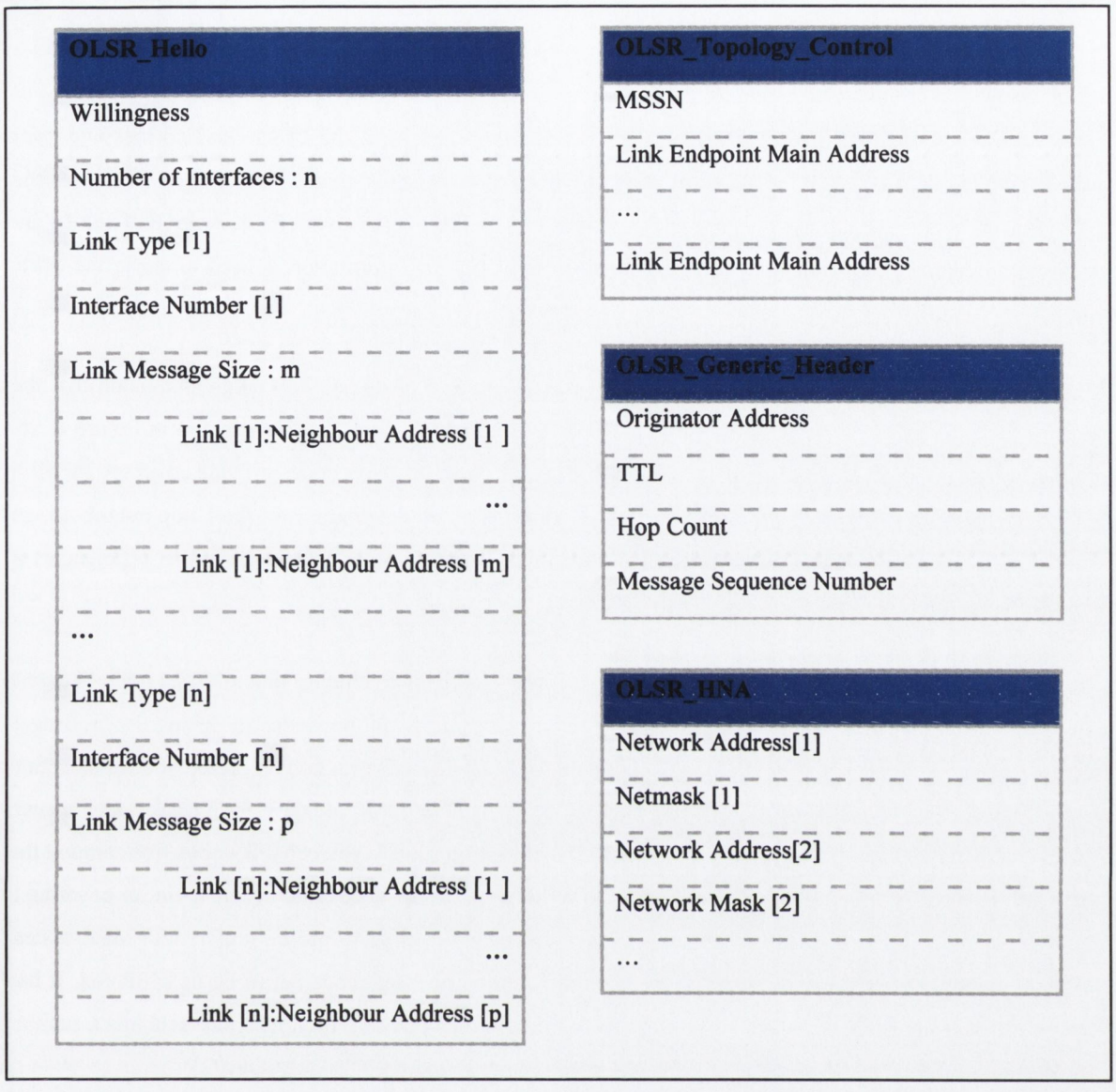


Figure A-8 OLSR Control Messages

Multipoint Relays:

Before describing the operation of OLSR, it is necessary to introduce the concept of multipoint relays. The multipoint relays are a key feature of this protocol as they are used to minimise the flooding of broadcast packets in the network by locally reducing the duplicate retransmission of such packets. Each node selects a set of its neighbourhood nodes which are in its *MPR set*. The difference between the neighbours of a node which are in its MPR set and those neighbours which are not is that only the MPR set neighbours retransmit packets that the node has sent to them. The other non-MPR neighbours only process packets from the node, they do not retransmit them. Neighbouring nodes must know if they are in the MPR set of a given node so that they will retransmit the broadcast packets as requested. To this end, each node keeps a list of its neighbours that are *MPR Selectors* of that node. A node is expected to retransmit a broadcast message that comes from one of the nodes in its MPR Selector Set. Nodes learn of their selection as MPR nodes through `OLSR_Hello` messages which are periodically beaconed.

When a node receives an `OLSR_Hello` message it updates its Neighbour Set, its 2-Hop Neighbour Set and its MPR Selector Table; these are among the state tables listed in Figure A.7. Each received `OLSR_Hello` message may contain a list of nodes that are neighbours of the node that originated the `OLSR_Hello` but that are not neighbours of the recipient node. For example, in Figure A.9a, the `OLSR_Hello` message from node 1 will declare nodes 2, 3, 6, 7, 9 and 10 as its neighbours (*the blue nodes*). When node 7 receives the message from node 1 it will only recognise nodes 2 and 3 as its neighbours. By default, nodes 6, 9 and 10 are its 2-hop neighbours. The Neighbour Set and 2-Hop Neighbour Set information is used to chose the MPR Set.

When each node selects its MPR Set it is aiming to pick a subset of its one-hop neighbours such that the union of their Neighbour Sets contains all of the node's 2-Hop Neighbour Set. To explain, in Figure A.9b, node 1 chooses nodes 2, 3, 9 and 10 as its MPR Set (*the pink nodes*)as these nodes provide coverage for all it 2-Hop Neighbour Set (*the yellow nodes*), i.e. each of node 1's 2-hop neighbours are the 1-hop neighbours of the selected MPR nodes. The MPR Set does not have to be optimal but the protocol loses its efficiencies if the MPR Set is not minimal.

The information in the `OLSR_Hello` message also contains a sequence number that allows a node to detect stale information and a Willingness indicator. The sequence number allows nodes to distinguish messages that have been delivered out of order. The Willingness indicator describes a node's preference for selection as a multipoint relay by other nodes. Relaying packets may be too resource intensive for a weak node, in such a case a node may set its Willingness level to zero. The algorithm used to select MPR nodes from among the symmetric nodes of the Neighbour Set may take the Willingness of the nodes into account. Nodes covering a larger subset of 2-hop neighbours and having a high Willingness setting would be preferential MPR nodes. As a non-optimal default, a node may choose its entire symmetric Neighbour Set to be its MPR Set. It has been shown that the MPR approach compares favourably to other flooding minimisation techniques such as Dominating Set Flooding, Reverse Path Forwarding and Super Flooding [Cla02c], [Qay02].

Topology Information Dissemination:

The efficient dissemination of the link state information in OLSR is built around the declaration of `OLSR_Topology_Control` (TC) messages that allow nodes to construct a routing table. TC messages are broadcast periodically throughout the network using the MPR system to minimise overhead. The TC message declares a node's MPR Selector Set, i.e. the message contains a list of the neighbours that have selected the originator of the TC message as their multipoint relay. The pink nodes depicted in Figure A.9b, nodes 2, 3, 9 and 10, would contain node 1 in each of their MPR Selector Sets as node 1 has chosen them to be in its MPR Set. Each node maintains a Topology Information Base that records information about the network that the node learns through the TC messages. Each entry in the Topology Information Base, a protocol state table listed in Figure A.7, consists of a destination node, i.e. an MPR Selector node as identified in the TC message, the address of the last-hop to that destination, i.e. the address of the originator of the TC message, and an associated sequence number that allows the node to flush out stale information. Each entry is deleted after a time if it is not refreshed.

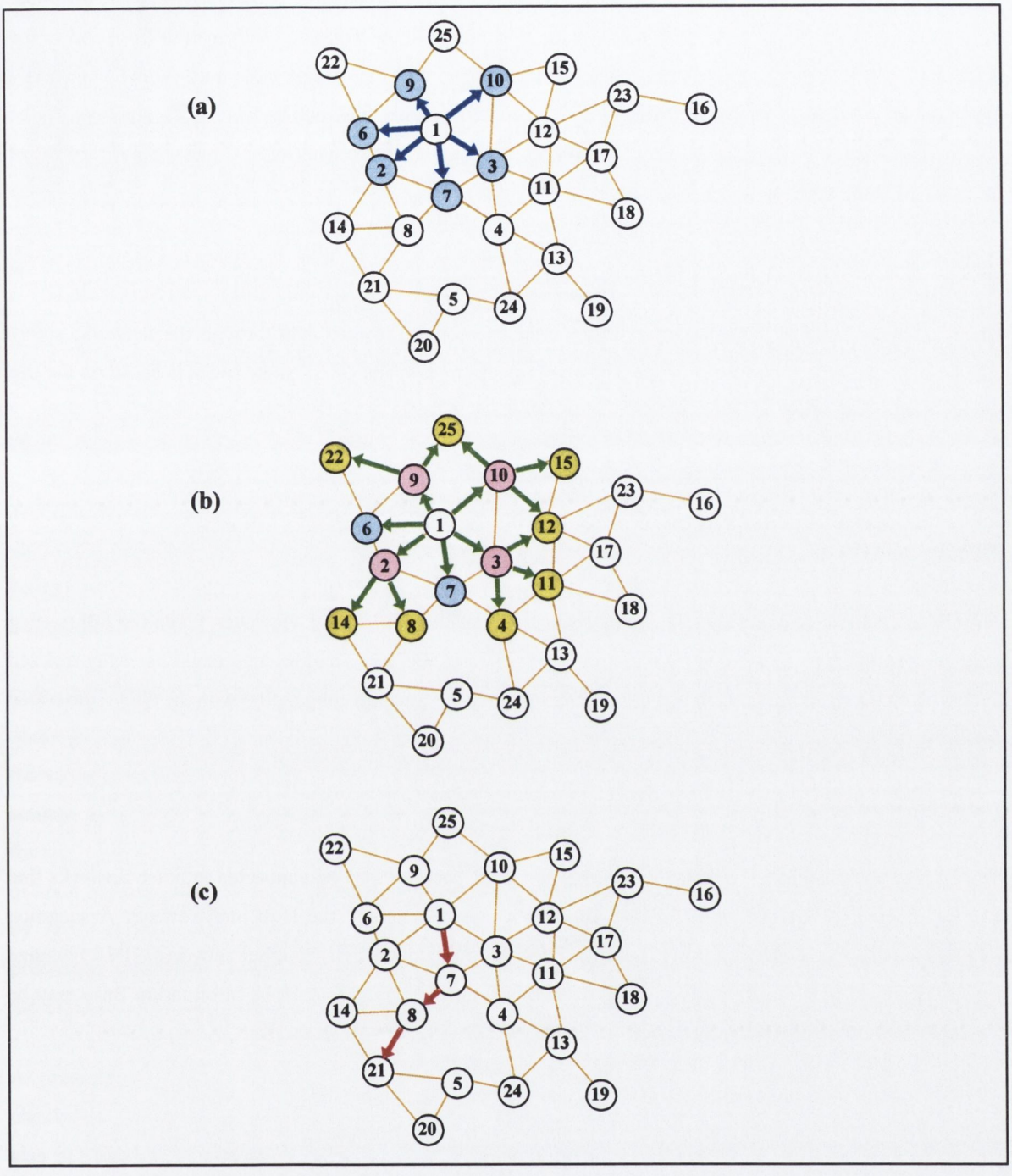


Figure A-9 OLSR Networking Example

Route Cache Management:

The final part of the OLSR protocol involves each node building a routing table based on the entries in its Topology Information base and its Neighbour Set. The information in the Topology Information Base is a list of pairs of nodes in the form *[last hop, destination]*. A node builds its routing table by going through this list and building up a route to each destination by linking the *[last hop, destination]* pairs together until the last hop is one of the entries in its Neighbour Set. For example, in Figure A.9c, a route from node 1 to node 21 may be established by linking the Topology Information Base entries *[last-hop: 7, destination: 8]* and *[last-hop: 8, destination: 21]* together with the Neighbour Set entry for node 7. The protocol employs a procedure that ensures that the shortest paths are calculated and then entered in the route cache for immediate use. In the

ongoing example, the route entry for destination node 21 would be *[destination: 21, next-hop: 7, distance: 3]*. As the routing table is based on the information contained in the Topology Information Base and in the Neighbour Set, if any change occurs in either of these nodes then the routing table must be recalculated. If there are no changes, i.e. if the topology does not change, then the routing table does not need to be recalculated. However, the OLSR protocol continues to issue Hello messages and TC messages regardless of the rate of change of the network topology.

Data Forwarding:

OLSR does not specify the mechanism that should be used to forward data through the network, solely concerning itself with the process of periodic topology discovery. The OLSR route cache is based on the use of link-state information.

■ **Additional Features**

Multicasting:

The OLSR protocol does not specify any mechanism to enable multicasting. However, given the full routing state information that is available from the protocol's normal operation, multicasting could be easily realised by use of simple tree construction and group membership algorithms in conjunction with the above-described protocol.

Heterogeneous Networks:

The OLSR specification recognises the fact that an OLSR network may be connected to other networks that are not running OLSR. The specification provides a message format, the Host and Network Association (HNA) message that allows 'gateway' nodes to advertise their connectivity to other networks. HNA message are disseminated in a similar manner to the Topology Control messages. A HNA Information Base may be constructed to hold the HNA information.

B Self-Stabilisation by Local Checking and Local Correction

B.1 Introduction

Self-stabilisation (or stabilisation) is the branch of computing that focuses on the ability of a system to converge, within a finite number of steps, from an arbitrary state to a state that exhibits desired system behaviour. In other words, a protocol would be said to be self-stabilising if its specification does not require a particular initial state to be imposed on the system to ensure the correct behaviour of the protocol.

Self-stabilisation by local checking and local correction is a technique that employs localised mechanisms to observe and correct the state of a network in a fast and robust manner. The technique can be used to transform an existing protocol into a self-stabilising protocol or to guide the design of new protocols from scratch. Correct network behaviour, reflecting the objectives of the network protocol is characterised in the form of a set of legitimate states, or *predicates*. The global state of a network is defined in terms of a number of such predicates. A predicate is simply a property or attribute of the network that can be affirmed or denied by some means, often taking the form of an evaluation function. In practical software coding terms, predicates will often take the form of *if/else* statements that may evaluate conditions (predicates) and control the execution of subsequent protocol steps.

A protocol is said to be stabilised when it reaches a global legal state. A network protocol is locally checkable if it has been designed such that whenever the protocol is in a bad state, some link subsystem is also in a bad state and that link subsystem can detect this fact locally. The global state of the network consists of the Cartesian product of the state of each node in the network and the state of the links intervening between each pair of neighbouring nodes, i.e. the messages in flight between each such neighbouring pair. In systems of interconnected networks, states are described by discrete variables and state transitions are described by transition rules, often in the form of programmes or protocols.

When the concept of local checking and correction is employed, each link subsystem is checked and corrected in parallel. The parallel checking and correction of the system ensures a fast recovery of the system when faults occur. Figure B.1 shows a single networking system of seven nodes which can be viewed as consisting of eight overlapping link subsystems. Link subsystem #5 consists of nodes u and v and the links between them. The local state of the (u, v) link subsystem consists of the 4-tuple of the state of node u, the state of node v, and the state of the two anti-symmetric links connecting them, i.e. the link from u to v and the link from v to u. This link subsystem will be used as an example throughout the following discussion.

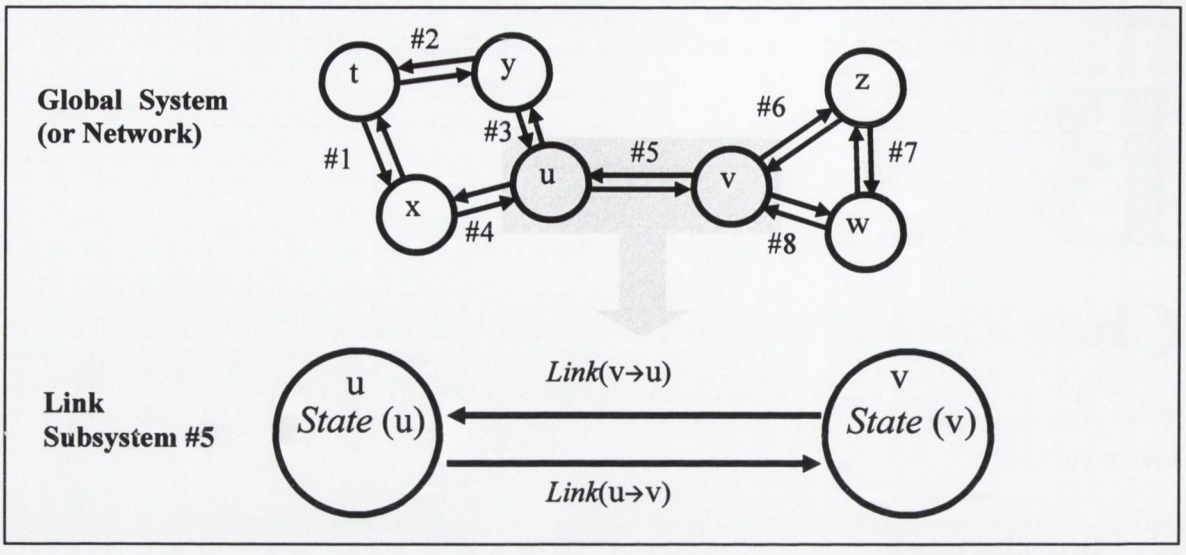


Figure B-1 One Global Network, Eight Local Link Subsystems

A *local predicate* of the global system (or network in this case) is defined in terms of a subset of the local states of that network. Specifically, a local predicate of the network for the (u, v) link subsystem is based on a subset of the states of the (u, v) link subsystem. A local predicate is defined such that only when the local link subsystem of the network, on which it is defined, is in a valid state should the local predicate be satisfied or hold true. A *link predicate* is the term used to describe the collection of local predicates, i.e. one for each link subsystem in the network. So, for the network depicted in Figure B.1, the link predicate would be the set of the eight local predicates for each of the eight link subsystems. Link predicates are defined such that only when the entire network is in a valid state should the link predicate, as manifested by the subset of states of each of the local link subsystems, be satisfied or hold true.

B.2 Local checking

The global state of a network protocol is locally checkable using a link predicate set if the global state can be represented by the conjunction of all the local predicates making up the link predicate set. Local checking is used to ascertain whether or not each local predicate holds. If a local predicate does not hold, then the link subsystem must be corrected so that the local predicate may be affirmed again.

Roughly speaking, the network depicted in Figure B.2 can be seen to be in an illegal state, with regard to the goals of the auto-configuration protocol, as one of its link subsystems is invalid. Link subsystem #1, which consists of Node 3 and Node 4, is in an illegal state as a DSR-configured node is connected to an AODV-configured node. This link subsystem violates the objective of the auto-configuration protocol; namely that neighbouring nodes operate the same protocol. The network is only in a legal global state when each of the link subsystems is correct, i.e. when every local predicate holds. On the other hand, link subsystem #2 is in accordance with the objective of the protocol as Nodes 6 and 7 are using the same protocol. If the invalid link subsystem was corrected by some means then the entire network would enter a globally legal state. The issues that arise in this situation are as to how nodes in the bad link subsystem detect the illegal state and how they correct the link subsystem. The detection of illegal predicates and the correction of such are not trivial.

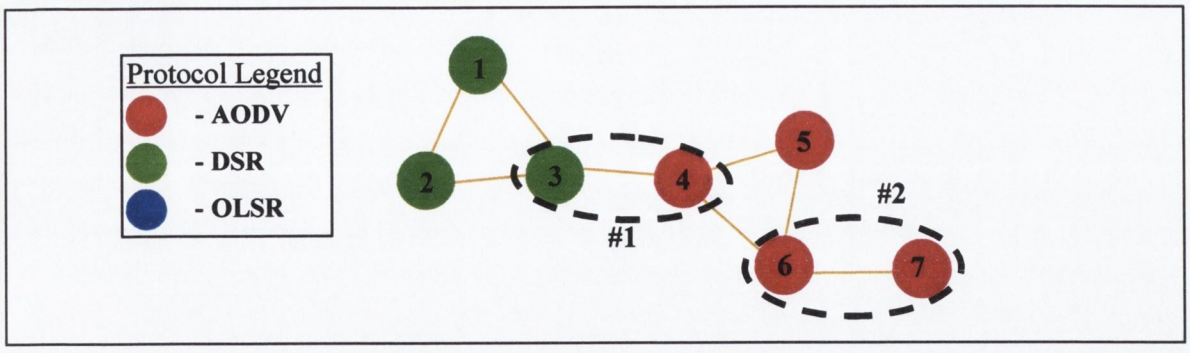


Figure B-2 A Network Featuring An Illegal Link Subsystem

For example, if Node 4 were to attempt to correct the subsystem by configuring itself to DSR, then link subsystem #1 would be corrected, but the other link subsystems overlapping at Node 4 would be invalidated. The correction of the invalid link subsystems must not invalidate another link subsystem which would further corrupt the global network state. To address such issues, the design of a locally checkable and correctable self-stabilising protocol demands that the local predicates be *closed* for local checking to be possible and that they be *stable* for local correction to be possible.

B.3 Closed Predicates

Local predicates are often two-way, i.e. they involve the state of node u, the state of node v, and the state of the links intervening between the two nodes, i.e. the link from u to v and the link from v to u. A local predicate of the network, which is defined by the state or *configuration* of the link subsystem, is closed if the predicate *still holds* when the state of the link subsystem has been subjected to a legal state transition by the network protocol controlling the configuration of the global system. Each time a node executes a step of the network protocol it may change the global state of the network; however, if the local predicate is closed then the global state cannot be invalidated by such a state transition. Basically, the protocol must include some rules that detect and exclude spurious data or messages that could send it into an arbitrary state, assuming it is initially in a correct state.

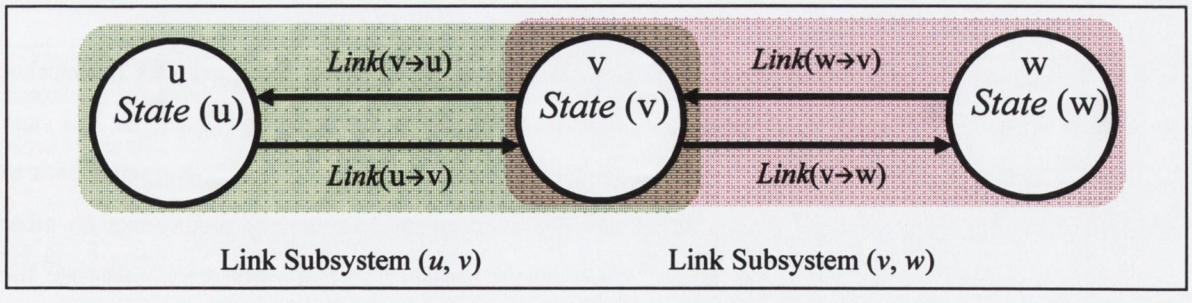


Figure B-3 Two Overlapping Link Subsystems

To clarify the purpose of local predicate closure, consider the system illustrated in Figure B.3 which consists of the two overlapping (u, v) and (v, w) link subsystems. The validity of the global state of this system is equivalent to the conjunction of the two local predicates, $L(u, v)$ and $L(v, w)$, where $L(u, v)$, for example, denotes the local predicate for the link subsystem which is dependent on the state of node u, the state of node v and the state of the links from u to v and v to u. It may occur that one local predicate in this system is

always false and that the false predicate is constantly changing from $L(u, v)$ to $L(v, w)$. As the system is asynchronous, the predicates cannot be checked all the time; they can only be checked on a periodic basis. Therefore, the checking procedure may never detect that there is a fault in the system as the fault continually moves from one link subsystem to the other. This is only possible if the protocol can be driven between illegal and legal states by arbitrary state transitions that are not controlled by the protocol. If the network protocol precludes the possibility of corruption, i.e. if the protocol detects bad values and ignores them, then state transition cannot be caused by spurious or corrupt data.

B.4 Two-Way Checking

If the local predicates are chosen such that they are closed, then the snapshot technique can be used to check the state of the local predicate, and thereby to check the global state of the network protocol. The snapshot technique generally employs a two-way mechanism whereby the leader of the link subsystem issues a Snapshot Request to the other node; leadership of a subsystem may be decided in an arbitrary manner, it may simply fall to the node with the higher unique node identifier.

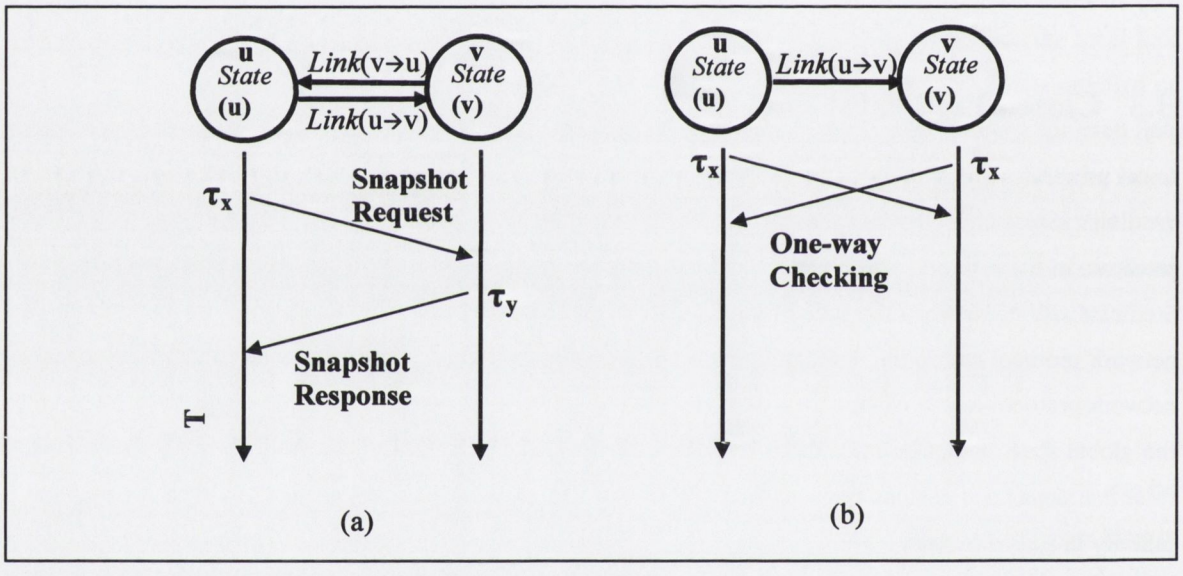


Figure B-4 Two-Way Snapshot and One-Way Checking

In the case of the link subsystem depicted in Figure B.4a, node u is the leader and it issues a Snapshot Request to node v at time τ_x. Node v records its state information in the received packet, i.e. the state information that is pertinent to the local predicate being checked, and sends the Snapshot Response back to node u at time τ_y. By way of example, a basic two-way snapshot mechanism may require that no other messages are sent across the link at the time of executing the snapshot, so that when node u receives the Snapshot Response from node v it will know the state of node u; its own state, the state of node v; which is contained in the Snapshot Response message, and the state of the intervening links; empty. Since the link predicate would be dependent on the state of that 4-tuple or a subset thereof, node u can readily check whether or not the link predicate holds. If not, node u can initiate the correction process for the link subsystem. More complex two-way snapshot techniques which allow for normal data flow to continue between the two nodes have also been developed. As noted earlier in the review section, the two-way

snapshot protocol itself has to be stabilised. As messages may be lost or delayed the snapshot mechanism must endure faults. However, this is easily overcome by the use of the counter flushing technique which requires that counters are added to each message so that the sending and receiving nodes may pair up Snapshot Request and Response messages, ignoring delayed and out-of-sequence messages. The counter flushing technique is already widely employed in the routing protocols in the form of *unique packet identifiers* and *sequence numbers*.

B.5 Separable Local Predicates and One-way Checking

It has been noted that the typical mobile wireless ad hoc network is susceptible to variable node-to-node link quality. Specifically, the links intervening between nodes may be asymmetric. This link characteristic would render the use of the two-way snapshot local checking technique unworkable. It is noted that the literature has identified an alternative and faster one-way snapshot technique as being suitable under certain imposed conditions; in order to use this technique the local predicate must be separable in addition to being closed. A separable local predicate is one that may be split into two one-way predicates; one one-way predicate for each link direction. The link predicate for the link subsystem illustrated in Figure B.1, which is dependent on the state of nodes u and v and the state of the two anti-symmetric links intervening between them, is split into two one-way predicates $O(u, v)$ and $O(v, u)$. The one-way predicates must be designed such that both one-way predicates must independently hold true for the local predicate to hold true. The one-way predicate $O(u, v)$ is checked by node u and is dependent on the state of local variables at node u and the state of node v as described by a message sent from node v to node u over the link. It should be keenly noted that the one-way predicate O(u, v) does *not* involve the state of the local variables at node v or the link from node u to node v. The other one-way predicate for the (u, v) link subsystem, $O(v, u)$, depends on the state of local variables at node v and the state of node u as described by the message sent from node u to node v over the link. In essence, the one-way predicate at node u, for any of its link subsystems, is only dependent on its own state variables and the messages from the other nodes in its various link subsystems regarding their state.

It has been shown in [Var92] that if the network protocol is one-way checkable, then it can be locally checked by periodic sending of state, i.e. if nodes u and v periodically send the pertinent state information to each other, then both nodes can check their one-way predicates for violations in lieu of using a two-way snapshot mechanism. As shown in Figure B.4b, nodes u and v can send state information to each other at time τ_x, rather than checking the link predicate using the slower two-phase approach depicted in Figure B.4a.

B.6 Local Correction

A network protocol is locally correctable if an arbitrary (or illegal) global state can be corrected to a desired legitimate global state by applying independent local actions, thus leading to fast and robust stabilisation. Local correction may be used to ensure that a local predicate is corrected when the local checking process detects a fault. The process should be as efficient as the checking process, otherwise the robustness and efficiency of the system, with regard to its self-stabilisation ability, is compromised. The primary literature on self-stabilisation by local checking and local correction [Awe91], [Var92], has proposed the use of a local reset mechanism to restore the local predicate at a link subsystem. Generally, the proposed local reset

technique has the aim of simultaneously applying a reset function to each of the nodes in the link subsystem, nodes *u* and *v*, whilst also clearing the intervening links of any messages; thus resulting in the state of the link subsystem being corrected in accordance with the link predicate. Although the simultaneous resetting of two nodes is not possible in an asynchronous subsystem, the literature proposes a two-way reset technique that has the same ultimate effect. When a local reset is applied to a link subsystem, the invalid global state of the network is transformed to another global state in which the local predicate holds again. In the literature, the specified local correction techniques actually involve initialising the values of the link subsystem, i.e. resetting the system.

Referring to the example of a credit-based flow control protocol described in Section 6.4.3 and illustrated in Figure 6.10, the Sender node could have initiated a reset by issuing a Reset Request message to the Receiver which would empty its queue upon receipt of the message, i.e. reset itself. The Receiver would then send the Reset Response back to the Sender which would then reset its credit register to the maximum value, thus stabilising the protocol. Again, such a two-way reset process may itself be stabilised by means of counter flushing. Alternatively, the Sender could have adjusted its credit register value upon receipt of the original Snapshot Response, i.e. the Sender could have *set* its credit register value to resolve the discrepancy between the two nodes rather than *resetting* both nodes by way of a request/response exchange. As with the one-way checking approach, this one-way correction approach would also be faster and more robust.

The literature defines a generic local reset function that will effect the local correction of a link subsystem. Such a local reset function would be applied at a node in the link subsystem, when a fault is detected, with respect to the other node in that link subsystem, i.e. node *u* would apply the reset function to itself with respect to node *v* so that node *u* enters a legal state, as dictated by the predicate that has been violated. In the case of a two-way reset, both node u and node v apply the reset function with regard to each other, thus stabilising the link subsystem. Alternatively, using the faster and more robust approach, one node sets its values to suit the other.

As may be expected, there are conditions to observe when designing the local reset function for local correction. Since overlapping subsystems may attempt to correct themselves at the same time, the correction of one subsystem may invalidate the correctness of another overlapping subsystem and may lead to so-called corrective thrashing between the overlapping subsystems. This would occur when subsystems are dependent on each other, and when the dependency is cyclic. Referring back to Figure B.3, if the (*u*, *v*) subsystem was dependent on the (v, w) subsystem, then the correction of the (*v*, *w*) subsystem may cause the (*u*, *v*) subsystem to become invalidated. However, if the subsystems can be designed to be independent of each other or if an acyclic dependency can be imposed on them, then the corrective thrashing can be avoided. Using the continuing example of the overlapping link subsystems depicted in Figure B.3, the stability property demands that the application of a local reset function at node *u* with respect to node v cannot affect the local predicate of the (*v*, *w*) link subsystem. An acyclic dependency relation can be assured by imposing a partial order on the link subsystems. If the link subsystem (*u*, *v*) is *less* than the (*v*, *w*) link subsystem, with respect to the partial order, then the time taken to correct the system is, at most, proportional to the length of the acyclic dependency chain.