

DECLARATION

I, Steven Silvio Pietrobon, based on my personal knowledge and information hereby declare:

1. I am presently the proprietor of Small World Communications in Adelaide, South Australia, Australia.
2. From 6 May 1997 to 5 May 2000 I was an Adjunct Research Fellow and from 6 May 2000 to 5 May 2015 I was a Senior Adjunct Research Fellow at the Institute for Telecommunications Research ("ITR") at the University of South Australia ("UniSA").
3. As part of my role as an Adjunct Research Fellow and Senior Adjunct Research Fellow, I maintained ITR's thesis webpage at <http://www.itr.unisa.edu.au/~steven/thesis/> ("Thesis Website").
4. I make this declaration based on my personal knowledge.
5. After a thesis was approved for publication by UniSA's Research Degrees Committee and ITR's Business Manager, my general practice was to upload the thesis to the Thesis Website.
6. On October 1, 1997 I contacted Peter Shoubridge to request a copy of his thesis to be placed on the Thesis Website upon final approval by the University. Exhibit A is a true and correct copy of that email.
7. On December 18, 1997 Peter Shoubridge emailed me a copy of his thesis entitled "Adaptive Strategies For Routing In Dynamic Networks" for inclusion on the Thesis Website. Exhibit B is a true and correct copy of Peter Shoubridge's email to me attaching a copy of his thesis.

EXHIBIT

Ex. 1006

8. On December 22, 1997 I received permission to post Peter Shoubridge's thesis to the Thesis Website from Dr. Graham Gurr, ITR's Business Manager. Exhibit C is a true and correct copy of that correspondence.
9. On or before January 6, 1998, I uploaded Peter Shoubridge's thesis entitled "Adaptive Strategies For Routing In Dynamic Networks" to the Thesis Website at the following location: <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>. As soon as the thesis was uploaded it became accessible to the public.
10. After uploading the thesis I emailed a link to the thesis to the ITR Thesis distribution list.
11. Exhibit D is a true and correct copy of an email dated January 7, 1998 replying to my email of January 6, 1998 to the ITR Thesis distribution list stating "Peter J. Shoubridge, 'Adaptive strategies for routing in dynamic networks,' Ph.D. Thesis, Uni. of South Australia, Dec. 1996. (613k)" is available at <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>.
12. Exhibit E is a true and correct copy of an email dated January 10, 1998 replying to my email of January 6, 1998 to the ITR Thesis distribution list stating "Peter J. Shoubridge, 'Adaptive strategies for routing in dynamic networks,' Ph.D. Thesis, Uni. of South Australia, Dec. 1996. (613k)" is available at <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>.
13. Because I do not presently have access to my directory on ITR's website, ITR's Director, Dr. Gottfried Lechner, has informed me that the file located at <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz> was last modified on December 18, 1997.

14. Exhibit F is a true and correct copy of Adaptive Strategies For Routing In Dynamic Networks provided to me by Peter J. Shoubridge, which has been publicly accessible since January 6, 1998 or before at <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>. Exhibit F was created on September 7, 2015 by downloading and unzipping pjs.ps.gz from <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz> and converting the PostScript file contained therein to a PDF file using Acrobat Distiller.

I declare under penalty of perjury under the laws of the United States of America that the foregoing is true and correct.

Sworn to this 8th day of September, 2015, in Adelaide, Australia

By: Steven Petrobon
Steven S. Petrobon

EXHIBIT A

From: Shoubridge, Peter <peter.shoubridge@dsto.defence.gov.au> on behalf of Shoubridge, Peter
Sent: Wednesday, October 01, 1997 11:16 AM
Subject: RE: Your thesis

Steven,

No worries.

Thanks,
Peter.

From: Steven Pietrobon
To: peter.shoubridge
Subject: Your thesis
Date: Wednesday, 1 October 1997 11:03

Dear Peter,

I maintain ITR's thesis web page at

<http://www.itr.unisa.edu.au/~steven/thesis/>

I understand that you are about to make final modifications to your thesis for final submission. When your thesis has been accepted I would greatly appreciate if you could send me a gzipped postscript file of your thesis so that I can put it on the web.

Yours sincerely,

Steven Pietrobon.

EXHIBIT B

From: Peter Shoubridge -CD 08-2596841 <Peter.Shoubridge@dsto.defence.gov.au>
Sent: Thursday, December 18, 1997 12:03 PM
Subject: PhD thesis for ITR Web site

Steven,

Arek Dadej has told me that my thesis has been approved by the Research Degrees Committee. Please find attached a gzipped postscript file of the thesis to place on the Web. Thank you.

Cheers and Merry Christmas,
Peter.

EXHIBIT C

From: Graham Gurr <Graham.Gurr@unisa.edu.au>
Sent: Monday, December 22, 1997 3:49 PM
Subject: RE: Peter Shoubridge's thesis

Steven, that will be fine. Thanks for your devotion to this task. Have a Merry Christmas.

Regards,
Graham Gurr

> -----
> From: Steven S. Pietrobon[SMTP:steven@sworld.com.au]
> Sent: Thursday, December 18, 1997 2:16 PM
> To: graham@spri.levels.unisa.edu.au
> Subject: Peter Shoubridge's thesis
>
> Graham,
>
> Is it OK to put Peter Shoubridge's PhD thesis "Adaptive strategies for
> routing in dynamic networks" (Arek Dadej supervisor) on the ITR thesis
> web site?
>
> Steven.
> --
> Steven S. Pietrobon, Small World Communications, 6 First Avenue
> Payneham South SA 5070, Australia fax +61 8 8332 3177
> <mailto:steven@sworld.com.au> <http://www.sworld.com.au/>
>

EXHIBIT D

From: augusto <lima@igs.net>
Sent: Wednesday, January 07, 1998 3:04 PM
Subject: Re: New Thesis

Dear Sir

Thank you very much for your information. I will make the download and read the thesis.
For your information my current study is PRMA - Packet Reservation Multi Access. Data and voice system.

Best regards

Jose Augusto de Lima

> From: Steven Pietrobon <steven@spri.levels.unisa.edu.au>
> To: steven@spri.levels.unisa.edu.au
> Subject: New Thesis
> Date: Tuesday, January 06, 1998 4:29 PM
>
> The following new thesis is now available:
>
> Peter J. Shoubridge, "Adaptive strategies for routing in dynamic
networks,"
> Ph.D. Thesis, Uni. of South Australia, Dec. 1996. (613k)
>
> <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>
>
> Steven Pietrobon.

EXHIBIT E

From: Tan Yong Boo/n <tanyb@mbox3.singnet.com.sg>
Sent: Saturday, January 10, 1998 7:03 AM
Subject: Re: New Thesis

Thanks.

Yong Boo

> From: Steven Pietrobon <steven@spri.levels.unisa.edu.au>
> To: steven@spri.levels.unisa.edu.au
> Subject: New Thesis
> Date: Tuesday, January 06, 1998 10:28 PM
>
> The following new thesis is now available:
>
> Peter J. Shoubridge, "Adaptive strategies for routing in dynamic
networks,"
> Ph.D. Thesis, Uni. of South Australia, Dec. 1996. (613k)
>
> <http://www.itr.unisa.edu.au/~steven/thesis/pjs.ps.gz>
>
> Steven Pietrobon.

EXHIBIT F

Adaptive Strategies For Routing In Dynamic Networks

Peter John Shoubridge

Bachelor of Engineering in Electronic Engineering

School of Physics and Electronic Systems Engineering

Faculty of Information Technology

University of South Australia

Thesis submitted for the degree of Doctor of Philosophy

December, 1996

Contents

1	Introduction	1
1.1	Thesis Objectives	2
1.2	Achievements	3
1.3	Thesis Outline	5
2	Cost of Routing in Networks	7
2.1	Routing in Communication Networks	7
2.2	Classifications of Routing Procedures	9
2.2.1	Deterministic and stochastic routing	10
2.2.2	Centralised and distributed routing	11
2.2.3	Hierarchical and nonhierarchical routing	11
2.2.4	Adaptive and nonadaptive routing	12
2.2.5	Collaborative and independent procedures	14
2.3	Shortest Path Routing	14
2.4	Flood Search Routing	20
2.5	Overheads Associated with Routing	22
2.5.1	Overheads in centralised and distributed routing	22
2.5.2	Overheads in hierarchical and nonhierarchical routing	24
2.5.3	Overheads in adaptive and nonadaptive routing	25
2.5.4	Routing table maintenance in adaptive schemes	26
2.5.5	Cost comparison of flooding and shortest path routing	29
2.6	Routing Cost and Network Capacity	30
3	Combining Routing Procedures	37
3.1	Future Demands on Routing	37

3.2	Background of Combined Routing Procedures	38
3.3	Constituent Protocols for Combined Routing Strategies	41
3.3.1	Distributed minimum hop algorithm	42
3.3.2	Jaffe-Moss algorithm implemented as minimum hop	46
3.3.3	Flood search algorithm	53
3.4	Multiple Protocol Routing	54
3.4.1	Operating multiple protocols in one network	54
3.4.2	Problems with alternating between routing protocols	58
3.5	The Proposed Hybrid Routing Scheme	60
4	Modelling of the Routing Strategy	64
4.1	Methodology	64
4.2	Development of the Simulation Model	67
4.2.1	Network and traffic models	68
4.2.2	Modelling dynamic network behaviour	72
4.2.3	Models of routing procedures	74
4.3	Comments on Interpretation of Simulation Results	77
5	Performance of Hybrid Routing	80
5.1	Performance of Minimum Hop and Flooding Protocols	81
5.1.1	Static network conditions	81
5.1.2	Verification of the simulation model under static conditions	85
5.1.3	Effects of dynamic network conditions	89
5.2	Implications of the Results to Multiple Protocol Routing	93
5.3	Performance of the Proposed Hybrid Routing Scheme	97
5.4	Effect of Network Size and Connectivity	102
6	Modifications to Hybrid Routing	108
6.1	Next Best Path Strategy	109
6.1.1	Delay throughput performance	110
6.1.2	Comparison with hybrid routing	112
6.2	Prioritised Routing Table Updates	115
6.2.1	The priority mechanism	115

6.2.2	Performance comparison using priority	117
7	Verifying Performance Robustness of Hybrid Routing	120
7.1	Expected Routing Algorithm Performance	120
7.2	Verification Through Simulation Modelling	121
7.3	Large Changes to Topology	122
7.3.1	Results of network simulation	123
7.4	Regional and Time Varying Link Cost Change Intensity	125
7.4.1	Simulation results	126
7.5	Topology Change in a Mobile Radio Network	129
7.5.1	Mobile packet radio node model	130
7.5.2	Topology control	131
7.5.3	Modelling a multiple access scheme	134
7.5.4	Routing performance	136
8	Conclusions	141
8.1	Identifying Dominant Routing Costs	141
8.2	A Solution to Routing Table Uncertainty	142
8.3	Recommendations for Further Study	143
	Bibliography	146
A	Routing Power Calculations	156
B	Dynamic Scenario for the Mobile Packet Radio Network Model	160
C	Publications	164

List of Figures

1.1	Routing performance subject to a changing topology	4
2.1	Shortest path spanning tree for destination d	17
2.2	Delay versus throughput for an M/M/1 queue	32
2.3	Delay versus throughput for network of queues	34
2.4	Relative performance of routing procedures	35
3.1	Location of node v in the network G	44
3.2	Routing overheads as the network becomes more dynamic	55
3.3	Multiple protocol routing	57
3.4	Primary node functions required for protocol switching	58
3.5	Local broadcast hybrid routing	62
4.1	Deriving a simulation model	66
4.2	Network model - 64 nodes with connectivity of degree 4	69
5.1	Delay using flooding and minimum hop in static network	82
5.2	Determining the maximum operating point, γ'_n	83
5.3	Queue length with flooding and minimum hop in static network . . .	84
5.4	Verification of minimum hop in static network	87
5.5	Verification of constrained flooding in static network	89
5.6	Delay using flooding with increasing intensity of change	90
5.7	Delay using minimum hop with increasing intensity of change	91
5.8	Dominance of retransmissions with increasing intensity of change . . .	92
5.9	Probability of success using minimum hop with increasing intensity of change	93
5.10	Delay using hybrid routing with increasing intensity of change	98

5.11	Relative routing transmission overheads	99
5.12	Comparative performance over varying intensity of change	101
5.13	Comparative performance - 64 nodes with connectivity of degree 8 . .	104
5.14	Hybrid routing performance with increased connectivity	105
5.15	Comparative performance - 128 nodes with connectivity of degree 4 .	105
5.16	Hybrid routing performance with increased network size	106
6.1	Delay using next choice with increasing intensity of change	111
6.2	Probability of success using next choice with increasing intensity of change	111
6.3	Relative routing transmission overheads	112
6.4	Comparative performance over varying intensity of change	114
6.5	Priority queueing model for packets transmitted on link l	116
6.6	Minimum hop routing with prioritised table updates	118
6.7	Hybrid routing with prioritised table updates	119
7.1	Minimum hop routing with increased degree of topology change . . .	123
7.2	Hybrid routing with increased degree of topology change	124
7.3	Comparative performance with increased degree of topology change .	125
7.4	Region of dynamic change (shaded area) moving across the network .	127
7.5	Minimum hop routing with regional topology change	127
7.6	Hybrid routing with regional topology change	128
7.7	Comparative performance with regional topology change	130
7.8	Delaunay triangles and some of their circumscribing circles	133
7.9	Minimum hop routing performance in the mobile radio network model	138
7.10	Flooding performance in the mobile radio network model	139
7.11	Hybrid routing performance in the mobile radio network model . . .	139
B.1	Stationary nodes and nodes travelling at 30km/hr	161
B.2	Nodes travelling at 60km/hr and 120km/hr	162
B.3	Delaunay triangulation of initial node positions	163

List of Tables

2.1	Routing Algorithm Performance	28
2.2	Relative Routing Cost	30
3.1	Distance and Routing Tables for Node v	44
6.1	Throughput and Delay Comparison with Hybrid Routing	113
6.2	Comparison of Minimum Hop Routing with and without Priority	119
6.3	Comparison of Hybrid Routing with and without Priority	119
7.1	Routing power performance in the mobile radio network model	140
A.1	Routing Power - 64 Nodes with Connectivity of Degree 4	157
A.2	Routing Power - 64 Nodes with Connectivity of Degree 8	157
A.3	Routing Power - 128 Nodes with Connectivity of Degree 4	158
A.4	Routing Power - Prioritised Routing Table Update Messages	158
A.5	Routing Power - 64 Nodes, 4 Degree Connectivity, $\mathcal{D} = 0.25$	159
A.6	Routing Power - 64 Nodes, 4 Degree Connectivity, Regional Change	159

Abbreviations

ACK	Acknowledgment
ARPANET	Advanced Research Projects Agency Network
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
AT&T	American Telephone & Telegraph
CDMA	Code Division Multiple Access
CSMA	Carrier Sense Multiple Access
CUP	Coordinated Update Procedure
DAR	Dynamic Alternative Routing
DNHR	Dynamic Nonhierarchical Routing
FDMA	Frequency Division Multiple Access
FLD	Flooding
IP	Internet Protocol
IUP	Independent Update Procedure
LAN	Local Area Network
MSG	Message
OSPF	Open Shortest Path First
RIP	Routing Information Protocol
SP	Shortest Path Routing
TDMA	Time Division Multiple Access
TSMR	Trunk Status Map Routing

Abstract

In modern communication systems, especially those with mobile radio components, the distribution of traffic loads and network topologies may vary from nearly static to very dynamic. The dynamic behaviour may vary both in space (i.e. affect only regions of a network) and in time. Commonly known routing algorithms tend to be well suited for specific networking environments. For example, algorithms based on the shortest path principle tend to work well with quasi-static networks, and random search or flood based algorithms are better suited to networks with dynamically changing topologies. As a result, it is very difficult to select a single routing algorithm that will be most appropriate for a given network, if the network is subject to varying degrees of dynamic behaviour. Usually, what works well for some regions of a network, will be inefficient for other regions. Similarly, the routing efficiency will vary in time. An intuitive solution to this problem, namely switching between a number of routing procedures depending on network behaviour, requires a set of switching criteria and decision making procedures which are extremely difficult to define and implement for networks with a distributed mode of operation and control.

This thesis develops a routing strategy that smoothly adapts to a changing network topology (defined by links failing and subsequently recovering) by combining two different routing principles into one hybrid routing procedure. The proposed routing procedure does not require any specific measures of the network dynamics to be computed by network nodes and then used to decide on the routing principle. Instead, by deciding locally on the basis of routing information normally available at each node, the way in which a next node is to be reached, the hybrid routing strategy exhibits a smooth change from minimum hop routing to constrained flooding, as the behaviour of the network (or regions within) changes from static to very dynamic.

Declaration

I declare that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text.

Peter John Shoubridge

Acknowledgments

I am very grateful for the support and guidance provided by several people in undertaking this research program. Firstly, I would like to thank Professor Michael Miller and Professor Ken Lever from the Institute for Telecommunications Research at the University of South Australia for their enthusiasm and the excellent postgraduate study environment they provided. Also, I wish to thank my employers, Mr Neil Bryans, Dr Tony Bedford, Dr Stephen Cook and Mr Manfred Heigl at the Defence Science and Technology Organisation for providing the opportunity and flexibility to undertake this study.

I wish to extend a very special thanks to my supervisor Dr Arek Dadej for the many constructive discussions we have had and his enthusiastic support that has been invaluable towards producing this thesis. Finally, I would like to thank Jonathon Shewchuk from the School of Computer Science at Carnegie Mellon University for permission to use his software to calculate the Delaunay triangulations used in Chapter 7.

Chapter 1

Introduction

Communication networks enable users to exchange information through the sharing of switching and transmission resources. Routing provides an essential function in ensuring that information transmitted from a source can successfully find a path through a network to its desired destination. Generally the routing function seeks to optimise some performance criteria such as network utilisation or throughput, or perhaps average delay. As networks increase in size, with more users demanding seamless connectivity, mobility and a high level of availability, the task of routing becomes an increasingly complex problem to network providers.

For efficient routing, nodes require at least some knowledge of network topology. Links may fail or become congested, nodes or users could be mobile, resulting in changes to the source-destination paths within the network. Information regarding these changes must be conveyed to network nodes so that appropriate routing can take place. This information exchange consumes network resources that could otherwise be used for user traffic and represents a cost in maintaining up to date routing information in network nodes. As topology or traffic loads change more frequently, the network becomes dynamic and maintaining accurate routing information at individual nodes comes at a higher cost.

In addition to transmission capacity, routing procedures consume other network resources such as processing time to execute algorithms and memory to store routing tables. Generally, in dynamic networks employing radio transmission bearers, processing power and memory within network nodes are of relatively low cost when compared to transmission resources which are often costly and limited.

1.1 Thesis Objectives

Typically, when shortest path routing algorithms¹ are utilised to maintain routing look up tables stored within network nodes, they contribute zero transmission costs when making a routing decision. In forwarding a user packet, a node simply determines the most appropriate outgoing link from the routing table. However, these shortest path algorithms do consume resources every time the network changes in bringing all the routing tables throughout the network up to date. If routing decisions are based on inaccurate information stored in routing tables, retransmissions or reattempts caused by higher layer end-to-end transport protocols typically occur as a result of the excessive delays or lost traffic. This places additional load on the network further consuming transmission resources.

Flooding based routing procedures, on the other hand, do not maintain routing tables and therefore do not require any table update procedures. As such they contribute zero update cost but do consume large amounts of network capacity through broadcasting each time a routing decision is required. Overheads resulting from these types of routing procedure are generally independent from the frequency of changes occurring within a network. Instead overheads are related to the extent of the flooding process.

While shortest path algorithms are less costly to operate in a reasonably static network environment, flooding may be required for very dynamic networks. This is certainly true if shortest path algorithms are ineffective in maintaining accurate routing tables due to high rates of change in network topology. Unfortunately flooding is inefficient, resulting in very low network utilisation. If network conditions can vary from static to very dynamic, better routing strategies are required for greater network utilisation. Identifying more efficient routing strategies under such conditions is the prime objective of this thesis.

¹Shortest path algorithms determine the “best” paths through a network as defined by arbitrary criteria. Full descriptions of such algorithms are presented in later chapters.

1.2 Achievements

With many routing algorithms operating efficiently for only one class of network, either nearly static or very dynamic, the task of selecting a single most appropriate routing algorithm can be very difficult because the dynamic nature of a given network can vary. Operating shortest path routing algorithms in dynamic networks is not possible due to the associated high routing cost. It has been established in this thesis that a dominant constraint limiting the use of minimum hop routing (one particular form of shortest path routing), in a network subjected to increasingly frequent link failures and subsequent recoveries, is the routing table uncertainty. While routing table update protocol overheads were found substantial, they are far less significant. This observation opens an opportunity for considerable improvement to the algorithm's performance by overcoming the problem of uncertainty in routing tables. It is possible to achieve significantly improved performance over a wider range of dynamic conditions by combining different existing routing procedures into a new hybrid routing procedure.

A hybrid routing strategy is proposed whereby a local broadcast, derived from a constrained flooding algorithm, is initiated if routing tables cannot provide a valid next node entry for forwarding user traffic. Under stable conditions the minimum hop algorithm provides a next node routing table entry for routing decisions. If this entry is unknown due to link failure, user traffic is broadcast on all available outgoing links (except the incoming link). Subsequent nodes further downtree towards the destination may use the minimum hop routing tables if a valid next node entry currently exists. This is possible because all nodes continue participating in the routing table update procedure. Once a node has restored a valid next node entry in its routing table, local broadcasting is no longer used. It is worth observing that the routing procedure defined here automatically adapts to changes in the network. Flooding occurs only for the parts of the network where routing tables are uncertain, and only for the periods of time when the uncertainty persists.

Figure 1.1 demonstrates the comparative network performance of flooding, minimum hop and hybrid routing as the network becomes more dynamic. Overall network performance is a function of average throughput and delay. Throughput is

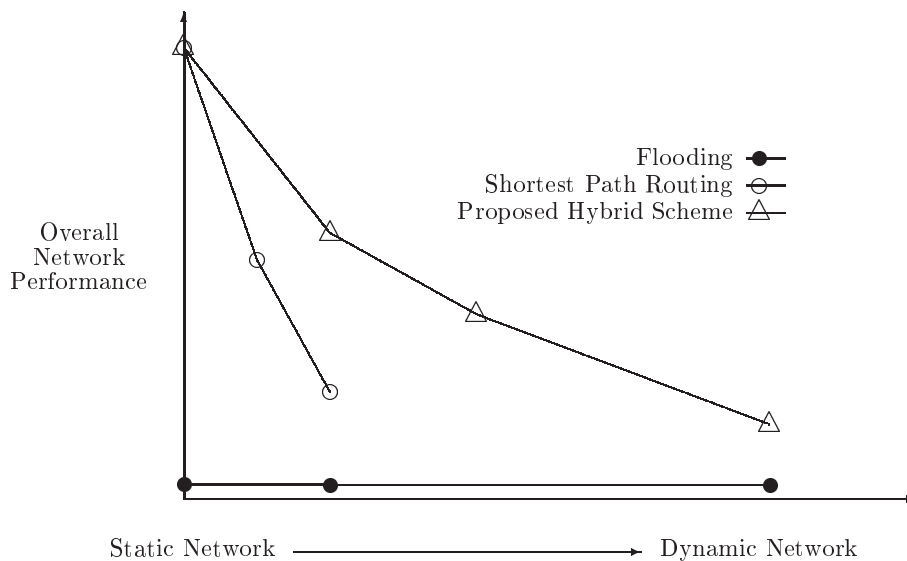


Figure 1.1: Routing performance subject to a changing topology

of concern to network providers interested in maximising infrastructure utilisation while delay represents the user perceived quality of service provided by the network. Dynamic behaviour of the network, such as changes in connectivity between network nodes caused by mobility or propagation effects, is modelled by randomly removing and replacing links in the network. The rate of these changes to the topology is increased to represent more dynamic conditions.

In this figure, flooding shows poor efficiency, but is also much more robust when compared with other routing procedures operating under dynamic network conditions. Minimum hop routing offers a vast improvement in performance when used in a static network but degrades as the network becomes more dynamic. As rates of topology change within the network increase, so does the extent of routing table uncertainty. This leads to a high probability of lost user traffic and reflects the algorithm's inability to track high rates of topology change. Minimum hop routing becomes inoperable under such dynamic conditions. However, hybrid routing offers superior network performance over a wide range of static through to dynamic network conditions. Operation with relatively high throughput under very dynamic conditions provides an indication of the algorithm's robustness, inherited from the flooding algorithm.

The essence of hybrid routing is to incorporate complementary features of generalised minimum hop and flooding algorithms into a more flexible routing strategy. This results in a relatively simple routing procedure that smoothly adapts from minimum hop routing in static and quasi-static conditions towards a flooding based procedure as the network becomes more dynamic.

A main contribution of this thesis has been the investigation of issues impacting on efficient operation of routing procedures in dynamic networks, leading to proposals for a new hybrid routing procedure for use in networks operating under static to very dynamic conditions. A performance evaluation of this strategy has been undertaken using simulation modelling techniques verified by queueing theory, and demonstrates that it leads to significant performance gains. Further enhancements to the hybrid routing procedure together with an assessment of its robustness have also been studied.

1.3 Thesis Outline

This thesis is structured in the following way. Chapter 2 introduces the problems associated with routing in dynamic networks. Principles of routing as well as various routing procedures are described with a focus on communication overheads and the implications of network capacity constraints. Chapter 3 describes in detail the specific routing algorithms investigated in this thesis and introduces the concepts of both routing protocol switching and hybrid routing as techniques used to reduce routing costs. The local broadcast hybrid routing procedure is presented in this chapter. Chapter 4 details the methodology used in the study of routing procedures together with reasoning behind the choice of this methodology. Specific details of simulation models are also described in this chapter. Reviews of previous work are included throughout Chapters 2, 3 and 4 as appropriate in the context of these chapters.

Performance evaluation of the proposed hybrid routing procedure is the subject of Chapter 5. Simulation model verification under static network conditions using analytical models is also demonstrated in this chapter. Additional mechanisms including the priority handling of control traffic to further improve network perfor-

mance are investigated in Chapter 6. Chapter 7 explores regional, correlated and other forms of non-random dynamic network behaviour to verify the robust nature of the hybrid routing procedure. Overall conclusions derived from the study are discussed in Chapter 8.

Chapter 2

Cost of Routing in Networks

Routing algorithms consume network resources in determining routes through a network to the required destination. In addition to overheads such as processing time for algorithm execution and memory to store routing tables, there is the consumption of transmission capacity that could otherwise be utilised for user traffic. Typically, computer processing power and memory requirements within network nodes are of low cost when compared with transmission overhead in networks using low to moderate link capacities. This chapter explores how various routing algorithms contribute to transmission routing costs and examines the associated impact on network performance.

2.1 Routing in Communication Networks

Routing can generally be defined as a network function that determines a path through the network from a source node to a given destination such that the path chosen optimises some criteria [1, 2, 3], [4, p. 259]. Example criteria could be to minimise delay, maximise network utilisation or perhaps minimise the use of expensive transmission bearers. The network itself consists of switching nodes interconnected by transmission links, with nodes redirecting user traffic along paths determined by the routing procedure. If routing tables are utilised, they are typically located within network nodes and used to decide over which link the user traffic should be forwarded towards a particular destination.

Circuit switched networks establish physical or logical connections between end

users before any user-to-user traffic is transmitted. Routing tables contained within nodes indicate the outgoing links to be used along a preferred path through the network. User traffic is then routed through the network along a selected path for the duration of the call or connection. It is the routing procedure's function to ensure that routing tables are updated and always reflect the optimum path through a network to be used by source-destination pairs at any given time.

Networks based on packet switching use a similar scheme known as virtual circuit routing to provide connection oriented services. As with circuit switching, a virtual connection or path is established at a connection set up time and all subsequent user packets are routed along this same path through the network. Routing tables contain entries mapping individual virtual connections with outgoing links for forwarding traffic to the specified successor nodes. As a result, user packets require a virtual circuit identifier in their packet headers so that a receiving node can determine the appropriate outgoing link from its routing table.

In addition to connection oriented services, packet switched networks can provide connectionless services using datagram routing. With datagram routing, each packet from a source node is switched through the network to its desired destination by individual nodes making routing decisions each time a packet is received [5]. As such, datagram packets are required to contain the full destination address in their packet headers. Furthermore, datagram routing can lead to subsequent packets taking different paths through the network, possibly causing packets to be received by the destination in a different order to the one in which they were transmitted by the source.

All the above network examples have in common the need for nodes to make routing decisions. These decisions may be based on fixed routing tables contained within each node, or the tables may be updated based on changes occurring within the network. Table updating requires the execution of a routing procedure to ensure that routing tables accurately reflect changes in the network. Alternatively, routing procedures may use stochastic rules and randomly select paths through the network, or simply broadcast traffic on all outgoing links to flood an entire network.

Before examining routing procedures themselves, let's consider how a desirable path through a network can be represented. A common approach in assessing the

paths through a network is based on the assignment of costs or weights to each individual link. Costs are assigned such that the global criteria to be optimised are reflected in a link cost function. Routing procedures can then use the measured link costs as a basis for calculating optimum paths through a network. A link cost can be based on fixed quantities such as link capacity, propagation delay, link security, or operation, maintenance and leasing costs of transmission assets. More dynamic or time varying link cost functions may include functions of traffic load measured as channel utilisation, buffer occupancy or average packet delay. Also, link error performance could be incorporated into the link cost function [4, p. 260], [6]. These static and dynamic parameters can be combined to provide a good assessment of how desirable a link is.

The average time interval over which dynamic link cost quantities are updated becomes an important parameter in dynamic networks. If the interval is too long, routing procedures may not be adaptive enough to changing conditions. An interval that is too short could result in instability due to drastic fluctuations in link cost or inaccurate measurements. Also, the synchronisation of link cost fluctuations can lead to instabilities [6, 7, 8]. These instabilities can be reduced by introducing a “hysteresis” property into the link cost function [4, p. 296]. By selecting an appropriate combination of static and dynamic parameters, the link cost function can be tailored for use by the routing procedure to meet the requirements of network providers in maximising utilisation while ensuring the user quality of service demands are met.

2.2 Classifications of Routing Procedures

Various taxonomies of routing procedures have been proposed, but it is difficult to include all existing routing strategies under any one of these taxonomies in a meaningful way. In general, literature describing the routing function and associated algorithms tends to illustrate routing concepts using specific algorithms to highlight the various properties. However, a selection of routing algorithm taxonomies will be presented in the following subsections to provide a framework for exploring complementary characteristics of algorithms and the evolution of some routing strategies.

2.2.1 Deterministic and stochastic routing

Routing algorithms have been classified as being deterministic or stochastic [9, p. 233], [10, p. 214], in a taxonomy originally proposed by Fultz [11]. Deterministic algorithms are static in that they do not adapt to changing network conditions but instead are based on averaged conditions. Fixed routing techniques where node routing tables are fixed for any given network configuration are deterministic. While these algorithms do not perform well in dynamic situations, the use of allowable alternate paths does improve survivability. The alternate path options are still fixed and hence deterministic. Shortest path algorithms can be used to calculate the best fixed routing rules for a given network. Alternate routing using fixed rules has typically been used in circuit switched networks [1, p. 8].

Another deterministic routing procedure is flood search routing or flooding. Flooding requires each node to route a received packet by broadcasting it on all outgoing links. This simple technique ensures that a packet will always get to its destination if a path exists in the network. Flooding is classed here as being deterministic because the routing rules do not change with variations in network conditions.

Stochastic routing is the proposed alternative class to deterministic routing. A simple stochastic routing technique is that of a random walk where a node forwards a packet by selecting an outgoing link at random. This type of algorithm is generally inefficient in terms of delay performance. Distributed stochastic algorithms that exchange information on observed traffic delay between neighbouring nodes in the network to favour lower delay links (by weighting link choice probabilities) can improve the routing performance. Alternative schemes where no information is exchanged are called isolated stochastic algorithms. Backward learning is an isolated stochastic algorithm whereby outgoing link choice is favoured by weighted probabilities towards the link offering lowest delay as measured from returning packets. Returning packets are those sourced by the destination node on transit to other nodes in the network. A performance comparison of deterministic and stochastic based routing rules is presented by Yum [12].

2.2.2 Centralised and distributed routing

A broad classification of packet switching routing algorithms into two classes, namely that of centralised and distributed algorithms, has been described by Rudin [13]. When routing strategies are prepared centrally and then routing information is disseminated to individual nodes for execution, the algorithm falls into the class of centralised schemes. Whenever the routing strategy is prepared throughout the network, distributed techniques are said to be used. Centralised techniques include fixed routing rules for a given configuration, rule updates provided by a central routing controller, and the ideal observer. The ideal observer has complete knowledge of the network at every instance in time and can be used to indicate the limiting performance of a network. Shortest path algorithms, where information is exchanged between neighbouring nodes and routes are calculated by individual nodes, fall into the class of distributed techniques. Other distributed algorithms are those of flooding, random and isolated routing as described previously. However, some routing algorithms such as Dijkstra's shortest path algorithm can be implemented as either centralised or distributed routing; this will be discussed in more detail later.

2.2.3 Hierarchical and nonhierarchical routing

In hierarchical networks, nodes are associated with a specific layer in a structured hierarchy. Global telephone networks typically used hierarchical routing due to its inherent simplicity in making routing decisions. Hierarchical routing evolved from the beginnings of automatic routing through direct and alternate routing schemes [14]. In a hierarchical network, a new call establishment procedure initially attempts the direct route to a destination peer node on the same hierarchical level. If this direct path is unavailable (perhaps due to blocking), an alternate route is attempted through another switch on the next higher level in the hierarchical structure of switches [15]. In this way traffic is always carried through the lowest available levels of the network hierarchy [1, p. 8]. Hierarchical routing uses simple fixed routing rules based on peak traffic loads in the network and as such these static rules are inflexible to changing traffic loads [4, p. 627].

Nonhierarchical dynamic routing appeared in telephone networks with develop-

ments in Stored Program Controlled switching and Common Channel Signalling systems enabling more flexible routing. Typically though, nonhierarchical circuit switched networks actually contain two levels of switching hierarchy, consisting of local and tandem or trunk exchanges. Local exchanges provide access to customer premises networks while trunk exchanges are used for accessing other regions of the network. These “flatter,” more highly connected topologies offer improved call blocking performance and networks are more survivable [16, p. 188].

Nonhierarchical time dependent dynamic routing has been implemented in networks such as AT&T’s Dynamic Nonhierarchical Routing (DNHR) system and uses fixed routing rules for a period of time, such as one hour. The DNHR scheme has improved network efficiency and reduced operational costs over hierarchical routing [17], it is also very effective in tracking real time changes in the network. DNHR establishes calls on a node-by-node basis and if a call is blocked at any particular node, the previous node involved in the call set up attempt tries an alternate route. This technique of alternate routing is known as crankback with the alternate routes chosen using fixed rules [16, p. 188].

Crankback may cause instability and network performance degradation if traffic load increases due to limiting the establishment of direct calls. This is caused whenever large amounts of overflow traffic prevent new direct calls being switched through to the designated alternate nodes. Trunk reservation on direct paths can limit the access of overflow calls and provide a protection mechanism against this instability [14]. Centralised adaptive routing techniques are used in the Trunk Status Map Routing (TSMR) system of DNHR to enhance performance by updating the overflow routes every 5 seconds, based on residual trunk capacity.

2.2.4 Adaptive and nonadaptive routing

Routing techniques have also been classified into adaptive and nonadaptive routing classes [18]. Nonadaptive routing is basically deterministic routing using fixed rules for a given network configuration. These rules are formulated in advance and the current state of the network is not taken into account. Adaptive algorithms, as the name suggests, adapt to changes in traffic patterns and topology. These schemes have been further classified into centralised, distributed and isolated classes. Cen-

tralised and distributed algorithms learn by communication with a central controller or neighbouring nodes respectively. Isolated schemes base routing decisions on information that individual nodes have each gathered independently without directly communicating with their neighbours, that is they learn by induction.

Adaptive distributed routing procedures have been proposed using techniques based on learning automata [19, pp. 362–379] and fuzzy set theory. Fuzzy set theory has been used to better interpret the uncertainty associated with routing decisions using distributed control strategies [20, 21].

Isolated adaptive routing procedures can readily be constructed using stochastic learning automata techniques because routing problems are typically well suited to decentralised control, where controllers are able to receive feedback that reflects the global performance criteria being optimised [22]. As a result, isolated adaptive learning approaches [23, 24] are increasingly being proposed for circuit switched telephone networks [25] and packet switched networks [26, 27, 28]. Routing algorithms of this type utilise feedback from information gained on network status and previous experience to modify decision rules on traffic routing. British Telecom developed Dynamic Alternative Routing (DAR) for their trunk network using a decentralised, isolated adaptive scheme based on learning automata. A technique called the “sticky principle”, that simply favours successful paths in the network, forms the basis of the DAR strategy [29]. Various other forms of isolated adaptive routing procedure have also been proposed [30, 31].

Most of the distributed adaptive shortest path routing algorithms, where route status information is explicitly exchanged between neighbouring nodes, have been implemented or proposed for packet switched networks [32, 33, 34, 35, 36]. These algorithms require a quasi-static network such that while network conditions can vary with time, some period of stability is required for routing tables to be updated. Distributed adaptive routing schemes tend not to be used in public circuit switched networks due to the high costs in making major nonstandard modifications to public carrier switch software [16, p. 185]. However, these schemes have been proposed for future connection oriented fast packet switched Asynchronous Transfer Mode (ATM) networks [37]. Many new issues regarding the various routing methodologies (including distributed shortest path and isolated learning schemes) for use in future

ATM networks are currently being investigated [38, p. 198].

2.2.5 Collaborative and independent procedures

Perhaps another taxonomy could be based on routing transmission overheads or costs, especially when considering the objectives of this thesis. Classifications could be based on whether nodes have prior knowledge of an appropriate path to a given destination at the time a routing decision is required. Shortest path algorithms are based on collaboration between neighbouring nodes, or central controllers, exchanging information to maintain routing tables for all destinations. This leads to high routing costs if many changes are occurring in the network. On the other hand, independent schemes such as flooding, shortest queue first or random searches require no knowledge of network topology and seek out the destination in isolation from other nodes within the network. These approaches introduce higher overheads in forwarding user traffic, but nodes operate independently from other nodes. However, as is the case with the other taxonomies described, it is not possible to classify all routing algorithms into one of these two categories, so this approach will not be pursued.

Before considering overheads associated with the various routing procedures outlined above, it is worthwhile examining some specific routing algorithms in more detail to better understand the issues involved. For this reason, shortest path routing and flooding are described in the following sections. These have been selected because they complement each other when operated in near static or very dynamic networks.

2.3 Shortest Path Routing

Consider a directed graph $G = (V, E)$ consisting of a finite set of V vertices representing nodes in a network and E edges representing the transmission links. The node set V has elements $\{1, 2, \dots, N\}$ representing all the nodes of the network while the set of edges, E , consists of links interconnecting the nodes represented as ordered pairs (i, j) of nodes such that $i, j \in V$. The edges are directed such that an edge (i, j) begins at node i and ends at node j . Weights can be assigned

to edges in the form of edge length, w , representing link costs in the network such that $w(i, j) \geq 0, \forall (i, j) \in E$. Let W be the set of all weights w_{ij} , where $(i, j) \in E$. Now the directed graph $G = (V, E, W)$ allows different link costs to be assigned to individual links in given directions.

Several distinct shortest path problems are defined but the unconstrained single source problem is the most common problem associated with network routing. In this class of problems no constraints are imposed, such as paths via specified nodes, and the shortest path from a given source node to every other node is sought. Many algorithms used to solve this class of shortest path problem have been proposed and studied [39].

The shortest path between a source node s and a destination node d is the train of edges between s and d with minimum length in G . The length of the directed path is given as the sum of all the edge lengths in the path between s and d . An intuitive way of solving this shortest path problem is to consider a physical model where the edges are pieces of string, each of length w_{ij} , corresponding to the respective edge weights connecting all the nodes together. By taking the source node s in one hand and the destination d in the other and pulling the nodes apart, the shortest path can be found as the train of edges pulled into a straight line between s and d [40, p. 162].

With increasing demand for distributed services and multipoint conference connections in communications, solving the shortest path subtree problem which contains a subset of nodes in V is also of interest [41]. Finding the shortest path tree connecting a given set of nodes in a graph is known as the Steiner tree problem. While this problem is NP-complete, various heuristics have been developed to solve the problem [38, p. 202]. The multipoint problem will not be considered any further in this thesis.

In most shortest path problems, edge lengths are fixed or assumed to be fixed for a period of time long enough for an algorithm to determine a solution, i.e. network conditions are said to be quasi-static. However, in some cases edge lengths may be treated as being time dependent, where edge lengths vary as functions of time, becoming the set $W = \{w_{ij}(t) \mid (i, j) \in E\}$. Solutions to shortest path problems with time dependent edge lengths have been investigated in [2, 42, 43].

These studies illustrate how predictions can be derived from time dependent changes through analysing patterns in change variation. Edge weights can also be considered as stochastic random variables [39] and often this is used to model uncertainty for network reliability analysis. Solutions overcoming the difficulties associated with incorporating probabilistic edge weights, representing link availability, into network routing algorithms are explored in [44].

The most efficient shortest path algorithm, operating under quasi-static conditions, is that proposed by Dijkstra [4, p. 268], [40, p. 103]. This algorithm requires complete knowledge of the network topology to determine the shortest path lengths from a source node to all destinations. Another algorithm, suited more to decentralised implementations, is that of Ford and Fulkerson which requires information on immediate neighbours only [40, p. 130], [4, p. 268], [45, p. 130]. Ford-Fulkerson's algorithm is also referred to as the Bellman-Ford algorithm because it is based on Bellman's equation [46].

Almost every shortest path routing algorithm implemented in current communication networks is based on either Dijkstra's or Ford-Fulkerson's algorithm [38, p. 198]. Using these algorithms, each node in a network generally maintains a routing table indicating the cost and next node (successor node) to a particular destination along the shortest path. As conditions in the network change, update messages are exchanged between neighbours or broadcast globally enabling nodes to recalculate optimum routing paths to all destinations. This results in the algorithm eventually converging to a stable state where no further changes occur to routing table entries.

Routing table entries for a particular destination in all nodes combine to form a shortest path spanning tree, rooted at that same destination. This is illustrated in Figure 2.1 for a single destination in an arbitrary network. Of course this is repeated for all destinations in the network resulting in a set of shortest path spanning trees. Distributed algorithms calculate these shortest path trees in a distributed fashion with nodes only containing information regarding their respective position in the tree. This is expressed in terms of the next node and cost to a particular destination.

Distributed algorithms have been used in the ARPANET to minimise delay [47]. In the original algorithm each node maintained a network delay table giving estimates for expected packet delay in reaching each destination. As delays change on

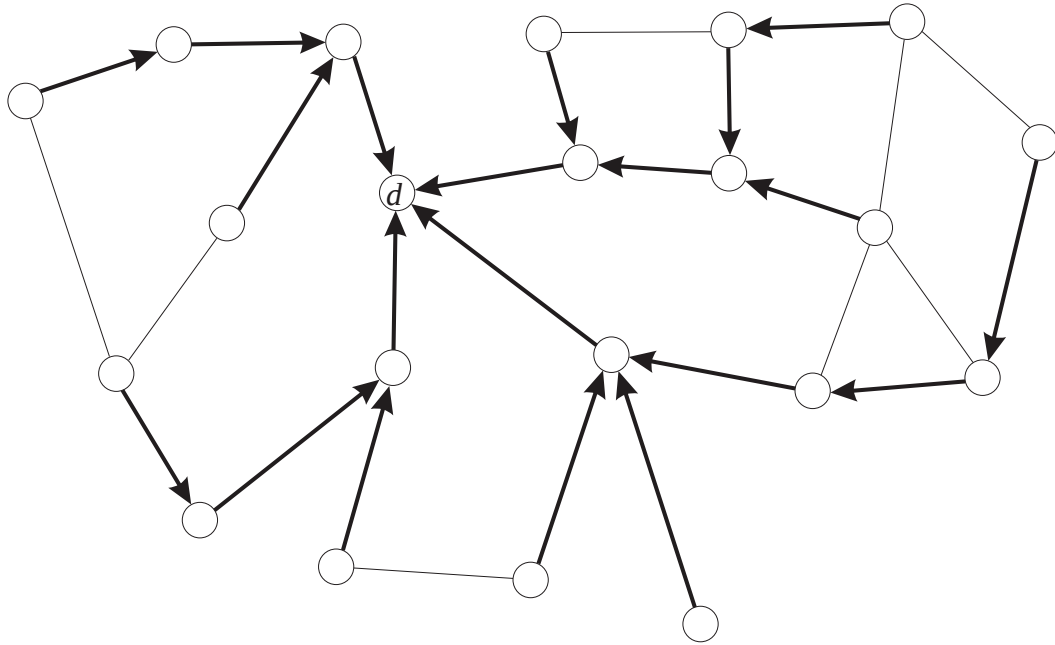


Figure 2.1: Shortest path spanning tree for destination d

various outgoing links, this table is updated and periodically the minimum delay tables are exchanged with neighbouring nodes. When a node receives a delay table from each neighbour it is able to update its own table with this additional information. This algorithm suffered from instability due to oscillations caused by delay tables being exchanged too frequently [34, p. 20].

A new algorithm was later developed, based on Dijkstra's algorithm, for calculating the shortest paths through the ARPANET [7]. In this algorithm, each node maintains a table describing the network topology and all the link delays. Shortest paths are calculated from this information and outgoing packets are routed along the shortest paths. The delays of each outgoing packet are measured for every link and the link delays are averaged over 10 second intervals. If a link delay changes significantly, the new delay is broadcast to all nodes in the network. A given node upon receiving these broadcast messages is then able to maintain its local delay table keeping it up to date with the status of global network topology and link delays.

Gallager proposed a minimum delay algorithm based on a static model that gives routes providing minimum overall delay of all packets transported through a

network [48]. This algorithm seeks to optimise global delay for the entire network whereas the ARPANET algorithms minimise delay for individual users. Merlin and Segall proposed an algorithm (following Gallager's approach) using distributed computation to ensure that loop free paths are established through the network [49]. The problem of packets looping back, and revisiting previous nodes, can arise in distributed techniques during update periods while a distributed algorithm converges to the true shortest path. Packet looping as described here is undesirable because it increases traffic load and packet delay in the network. Under certain conditions, such as link failure occurring in a chain of nodes, nodes uptree from the destination can become isolated. This results in packet looping because nodes exchange new distance values, incremented by a link cost unit each cycle, unaware that the destination is actually isolated. This is often referred to as the counting-to-infinity problem which is evident if no upper finite bound is placed on the distance values [4, p. 316].

In addition to being loop free, the Merlin-Segall algorithm is also event driven with each destination triggering the distributed updates for all routes to that destination. Formal description and validation of distributed network protocols, including the Merlin-Segall algorithm, have been published [50, 51]. Another distributed algorithm using link change events to trigger an update protocol, rather than using periodic updates, is described and proven correct by Tajbnapis [35].

Jaffe and Moss proposed a distributed shortest path algorithm offering improved performance in speed of convergence and ensuring loop free paths after a change in topology [52]. This algorithm consisted of two procedures, an Independent Update Procedure (IUP) and a Coordinated Update Procedure (CUP). The IUP is based on a distributed version of Ford-Fulkerson's algorithm operating in the presence of static or decreasing link costs. Jaffe and Moss determined that link cost increases cause loops to form during independent update procedures and developed the CUP to prevent the formation of these loops. This routing procedure has been selected as the shortest path algorithm to be used for further investigations in this thesis. A detailed description of this routing procedure is given in Chapter 3.

Update coordination required in algorithms such as Merlin-Segall or Jaffe-Moss inhibits the ability of a node to rapidly obtain alternate paths in large networks subject to frequent topological change. A comparison of the Merlin-Segall algorithm

and a distributed version of Ford-Fulkerson's algorithm (which does not employ any update coordination) used in a satellite network illustrates the performance degradation caused by coordinated update procedures [53].

An efficient minimum hop algorithm in which link costs are fixed to one has been proposed by Garcia-Luna-Aceves [54], with improved performance over previous loop free algorithms. This new minimum hop algorithm avoids any coordination of update procedures and retains the simplicity of the Ford-Fulkerson distributed algorithm. The problems of loop formation and counting-to-infinity as described earlier are avoided by nodes exchanging more than just distance information. Nodes also exchange next node information and this is used to ensure that routing paths are chosen based on either constant or decreasing distances and constant node constituency. So by using more than strictly local information, loops can be avoided. A constraint of this algorithm is that it requires link costs to be of unit or infinite cost only, however minimum hop routing is often used in packet radio networks making this algorithm attractive. It is claimed that this algorithm can be extended to the more general case of least cost rather than minimum hop routing.

Shortest path routing is used to configure routing tables within Internet Protocol (IP) gateway nodes for the interconnection of Local Area Networks (LANs) forming the Internet. Typically, broadcast routing protocols are used within the individual LAN domains. Current Internet routing is performed by the Routing Information Protocol (RIP) based on a distributed version of Ford-Fulkerson's algorithm [55, 56], [57, pp. 65–98]. Future Internet routing is expected to use the Open Shortest Path First (OSPF) protocol. This protocol supports clustering of nodes into a hierarchical network structure consisting of nonhierarchical subnetworks grouped into Areas which are themselves interconnected through higher level Area Border nodes [57, p. 115]. The OSPF routing procedure is based on a decentralised implementation of Dijkstra's algorithm with a flooding protocol used to update topology databases [57, pp. 99–134] and assumes networks are near static [58].

Generally, the possible formation of loops, and subsequent need for coordinated update algorithms, is not required due to the reasonably static nature of corporate networks. However, this assumption is not expected to be valid in future networks requiring routing procedures that are able to support rapidly changing topologies,

such as wireless computer networks [59]. Topology control strategies have also been proposed and modelled for the dynamic insertion and removal of links within a wide area packet/router network to improve network utilisation and reduce congestion [60].

2.4 Flood Search Routing

Flood search or flooding techniques are most useful in very dynamic networks where destination nodes are highly mobile or topological change occurs very frequently and network connectivity is uncertain. Flooding algorithms simply flood an entire network or subnetwork with packets until the desired destination is reached. While flooding is very wasteful of network resources, due to the large number of duplicated packets, it is extremely robust because all paths to the destination are attempted.

Connectionless datagram services can readily be supported using flooding by individual nodes broadcasting user packets through the network until the destination is reached. No routing tables are required. Alternatively, flooding can be used to determine the path to a given destination for the establishment of a virtual circuit connection. A source node floods the network with each packet recording its own path through the network. The destination, on receiving the first packet originated from the source, returns an acknowledgment along the same path back to the source node. Intermediate nodes along this path then configure routing tables for forwarding user traffic associated with the virtual connection. Once the connection is established, user traffic can be routed via this path through the network for the duration of the call. This procedure of flood-and-forward routing is relatively unreliable when compared with general flooding schemes [61]. It is though much more efficient because only an initial packet is flooded for the purpose of connection establishment.

A technique called controlled flooding can be applied to limit the extent to which packets are flooded through a network. With this routing procedure, packets are given a wealth value by a source node and each link in the network possesses a link cost. As packets traverse a link during the flooding process, their wealth is decreased by the associated link cost. Once a packet has insufficient wealth to traverse any

further links, the flooding stops. By adjusting the amount of wealth assigned at a source node, the degree of flooding can be controlled. It is possible with high enough wealth to flood the entire network. Techniques have been investigated that optimise performance by assigning fixed wealth to packets at the source and allowing transit nodes to adjust their link costs [62]. This could be particularly useful for virtual circuit routing where feedback is obtained on the current distance to destination nodes.

Constrained flooding is another extension to basic flooding where packets associated with a single flood search contain a unique sequence number. Nodes only broadcast received packets with the same sequence number once. Duplicated packets arriving at a node with the same sequence number as a previously flooded packet are discarded. To implement this, each node requires a table to store sequence numbers of all packets previously broadcast. Sequence numbers of previously flooded packets must be retained for at least the duration of a broadcast within each node [61].

A modified version of flooding has been proposed for tactical military circuit switched networks [63]. The robustness of flooding that guarantees finding the path to a destination, if one exists, is required for the high level of personal mobility and priority needs of tactical network subscribers. If a subscriber wishes to call another subscriber, not affiliated with the same network node, “request for routing” messages are broadcast to all adjacent nodes. If the called party is not affiliated with any of these nodes the flood search continues. Once the desired destination node is found, a circuit switched call can then be established. This approach is similar to the flood-and-forward routing principle described previously.

Another algorithm proposed for packet radio networks uses a technique called directed flooding [64]. In directed flooding, the objective is to connect a mobile radio node to a fixed network infrastructure. Flooding is used, but nodes only broadcast packets towards the fixed infrastructure to reduce the overhead of duplicate packets. Once in the fixed infrastructure, shortest path algorithms can be utilised.

2.5 Overheads Associated with Routing

To perform the routing function, network resources are consumed by routing algorithms and the use of these resources can be regarded as a cost. This is because the network resources consumed can no longer be utilised for user traffic. Network resources consumed are dependent on, computational complexity of routing algorithms, storage requirements within network nodes and the transmission capacity used in exchanging routing information between these nodes (or a central controller). This section explores how various routing strategies consume these network resources following the same framework of algorithm taxonomies presented in Section 2.2.

Computer processing power and memory are becoming cheaper, however processing requirements within network nodes are increasing. This is due to the increased complexity in managing and controlling networks supporting integrated traffic types (voice, video and data) and value added intelligent network services.¹ Advances in transmission technology are continually providing increased capacity over transmission bearers at lower costs. However, requirements for transmission bandwidth increase as the number of users increase, and as users demand higher bandwidth applications with greater capabilities for information exchange and sharing. Since the cost to procure, operate and maintain transmission infrastructure is still significant, and the bandwidth demands are ever increasing, transmission capacity remains a costly resource to be utilised efficiently. This is especially true in networks utilising satellite and terrestrial radio transmission bearers where transmission capacity is severely limited.

2.5.1 Overheads in centralised and distributed routing

In a centralised routing scheme, routing calculations are performed by a central network controller with the resulting routing information disseminated to network nodes. Global knowledge of the network topology is stored within the central controller with nodes informing this controller of changes in network status or topology. Storage requirements for routing tables will be very large with long delays in updating these tables, particularly in large networks responding to change [8, 37].

¹Such services include broadcast, multiparty conferencing, interactive and multimedia services.

Network survivability is reduced with a vulnerable central point of failure [65, 66]. The combination of high overhead and low survivability makes centralised routing procedures unattractive.

Shortest path routing procedures based on Dijkstra's algorithm are ideally suited to centralised implementations because this algorithm requires global network knowledge. Furthermore, by ensuring that every node in the network possesses complete global knowledge of the topology, an algorithm such as Dijkstra's can be implemented as a distributed routing procedure. This approach has been adopted in the ARPANET with an update protocol being used to broadcast topology change information throughout the network [54, 7]. This enables improved survivability because the routing decision making process is now distributed. However, information regarding every change in a network's topology must now be transmitted to all nodes in the network, resulting in high overheads if the network is very dynamic.

Decentralised or distributed routing schemes in general rely on local knowledge from neighbouring nodes and adjacent link status to calculate the routing rules. This substantially reduces the overheads associated with updating nodes with network topology information and nodes are able to respond quickly to local changes. Distributed shortest path algorithms using local knowledge can readily be derived from Ford-Fulkerson's algorithm and are relatively simple to implement. Since overheads associated with topology information updates are reduced because nodes only exchange information with their neighbours, these routing procedures are suited to large networks with dynamic topologies [54].

A combination of centralised and distributed routing in which responsibility for routing decisions is split between a central controller and individual nodes offers a compromise in overhead and performance. An algorithm called Delta Routing takes advantage of global updates from a central controller for accurate information and uses isolated routing within network nodes to rapidly adapt to local changes [13].

2.5.2 Overheads in hierarchical and nonhierarchical routing

In general, hierarchical routing is simpler to implement than nonhierarchical routing but exhibits reduced performance. A hierarchical network operates with nodes only requiring knowledge of their respective peer nodes' subnetwork topology. Any routing decisions involving nodes outside a peer domain requires routing via a higher level node possessing a broader perspective in its view of the topology.

Nonhierarchical routing treats all nodes as peers within the entire network. Routing tables at each node maintain lists of connections to destination nodes that can be referenced during call establishment. As conditions change in the network these tables can be modified. As such nonhierarchical routing requires more complex processing and larger storage capacity within network nodes. Additionally, a signalling capability is required to communicate with other nodes in the network for the purpose of exchanging routing information; this consumes network bandwidth. Benefits of nonhierarchical routing include the ability to adapt to changes in traffic patterns and cope with network failures, improving availability. This technique can greatly improve user quality of service at reduced operating costs [1, p. 51] by allowing increased throughput for the same call blocking probability when compared with hierarchical routing in telephony networks [30, 67].

A flat network where all nodes and links possess equal importance provides the most survivable topology. Alternatively, a hierarchical network contains critical nodes and links that if removed, partition the network. Measures of survivability given to a network based on the content of critical nodes and links in a topology have been proposed [68]. Generally, the benefits from using nonhierarchical routing outweigh its disadvantage, namely the consumption of network resources.

Performance gains from nonhierarchical routing decrease as network size increases due to the high overheads in maintaining routing tables. Some studies indicate that this occurs in networks with greater than 100 or 200 numbers of nodes [69]. A form of hierarchical structure is therefore required to overcome this problem and enable efficient routing in large networks. Various studies have investigated the concept of clustering nodes to form nonhierarchical subnetworks linked together

by higher level nodes forming a hierarchical structure [34, 69, 70]. The routing algorithms operating within node clusters are essentially nonhierarchical. Typically, the computational complexity of nonhierarchical shortest path algorithms is $O(N^3)$, and the size of routing tables $O(N)$, where N is the number of nodes ($O(N)$ means “order of” N). Creating a hierarchical network using clusters of nodes to form domains can realise performance gains with computation of complexity $O(\sqrt{N^3})$, and memory complexity $O(\sqrt{N})$ [70]. These approaches essentially utilise the combined advantages of hierarchical and nonhierarchical routing for use in very large networks.

2.5.3 Overheads in adaptive and nonadaptive routing

Consider the network resources consumed in maintaining routing information at individual nodes when decentralised, nonhierarchical routing procedures are used in large dynamic networks. These algorithms rely on routing look up tables and as such contribute zero transmission cost in making a routing decision to forward packets towards a successor node. They do however consume resources every time the network topology changes in order to bring all routing tables throughout the network up to date. These updates can occur periodically or after an event, such as a node or link failure.

Dynamic networks such as packet radio networks with highly mobile nodes can generate excessive quantities of control traffic in executing distributed algorithms for updating routing information [71]. Uncertainty in the accuracy of routing tables in such networks can lead to long periods of instability resulting in congestion and degraded network performance if adaptive routing schemes are used [72]. In mobile networks that are rapidly deployed, such as military tactical networks, simply establishing the initial routing tables in a reasonable period of time is itself a very difficult task [73]. However, using flooding in such a network can also cause problems due to collisions between simultaneous broadcasts throughout the network; this is in addition to the high overhead [74].

Isolated adaptive algorithms where no information is exchanged between neighbours (instead they learn from local experience) are naturally more efficient in terms of topology update overheads but tend to suffer from degraded performance. In networks such as congested and damaged circuit switched networks, it has been

observed that isolated adaptive learning schemes can perform well with node damage but deteriorate network performance under congestion. No learning is actually desired in cases of short term congestion [16, p. 233].

By allowing limited communication between neighbouring nodes, in addition to local learning abilities, overall performance can be improved [18]. A distributed failure detection algorithm that learns to keep calls destined to damaged nodes out of a circuit switched network can be almost as effective as adaptive routing with global knowledge [75]. This technique is more efficient than communicating topology update information and releases resources for use during network restoration.

Isolated adaptive routing procedures can be expected to suffer in performance when used in dynamic networks due to the relatively slow diffusion rate of information through the network. Each node can only learn indirectly from other nodes in the network and in a dynamic network the diffusion rate may be too slow to be of any use in a global sense. A coherence length, defined as the distance at which information becomes unreliable relative to the frequency of change in network status, increases as the rate of information diffusion through the network increases [16, p. 216]. If the coherence length is not equal to at least the network diameter then inaccurate information is most likely contained within routing tables.

2.5.4 Routing table maintenance in adaptive schemes

All shortest path algorithms will provide identical routing tables for a given topology. How well these algorithms adapt to changes in network topology or load conditions indicates the algorithms' performance. Performance can be assessed by criteria such as speed of response, number of control packets transmitted, size of control packets, computational complexity, memory space requirements and the possibility of loop formation [4, p. 314]. Computational complexity and memory requirements within nodes become more important as network size and the number of users increase. A concept of layering routing functions has been proposed to overcome this [37].

Speed of response is determined by the time taken for information regarding a change to propagate throughout the network so that all routing tables can be updated to the new set of shortest paths. This settling time required to reach a stable state is also referred to as the algorithm's convergence time. Convergence

time is an important criterion for routing algorithms because during an update cycle some routing tables within the network inevitably contain inaccurate routing information. Furthermore, the convergence time of an algorithm must be faster than the rate at which the network topology changes or else the algorithm will never converge and routing tables will always be inaccurate. As such, shortest path algorithms demand some period of stability after change to allow this convergence. The result is near optimum routing for most of the time, if networks behave in a quasi-static manner.

Another important criteria are the number and size of control packets transmitted during an update cycle. This control traffic consumes network resources and limits available capacity for user traffic. The amount of control traffic generated can vary greatly depending on where the changes occur [76, 77], [4, p. 326].

If during an update cycle loops form in the network routing tables, due to outdated network information, user traffic may be trapped in loops for a period of time until tables are updated, experiencing delays and wasting network resources. Loop formation can also result in packets being received in a different order to which they were sent, making these algorithms unsuitable for virtual circuit routing. Loop free routing during update cycles is therefore very desirable and a key feature of recent generation routing algorithms.

A final issue that remains to be considered is the overall cost associated with operating shortest path algorithms in very dynamic networks, due to the combined effects of high update traffic loads and routing table uncertainty. Table 2.1 compares the convergence time to determine a new shortest path (in number of update cycles) and the number of update control messages exchanged between neighbouring nodes during algorithm convergence, for a selection of algorithms. Results presented in the table were obtained from [52, 54, 78]. Where N denotes the number of nodes, L is the number of links and D the average degree of connectivity. Diameter, r , is the longest of all shortest paths between any two nodes in the network, and the average number of nodes affected by a cost change event is x . The performance notation of “order of,” $O(\cdot)$, is again used due to the wide variations observed in these parameters for different cost change events. Large variances observed in simulation experiments for the number of iterations and update message counts are examined in [78] and [4,

Table 2.1: Routing Algorithm Performance

Routing Algorithm	Convergence Time	Update Messages
Ford-Fulkerson (cost increase)	$O(N)$	$O(N^2)$
Ford-Fulkerson (cost decrease)	$O(r)$	$O(L)$
Merlin-Segall	$O(r^2)$	$O(N^2)$
Jaffe-Moss	$O(x)$	–
DUAL	$O(x)$	$O(6Dx)$

p. 326].

Typical first generation distributed shortest path algorithms, such as Ford and Fulkerson’s algorithm, suffered from the counting-to-infinity problem whenever a cost increase resulted in any destination becoming unreachable. This deterioration in performance after a cost increase is shown in Table 2.1. The counting-to-infinity problem causes high update traffic loads and longer convergence times, it is also the main cause of packet looping. This algorithm is simple to implement and suited to large, relatively static networks. Loop free algorithms that counter this problem using some form of coordination show significant improvement in performance, see Merlin-Segall, Jaffe-Moss and Diffusing Update Algorithm (DUAL) in Table 2.1.

While coordination in the Jaffe-Moss algorithm prevents loop formation, it reduces the nodes ability to obtain alternate paths, particularly in large networks with frequent topological changes [54]. Even when using this algorithm’s Coordinated Update Procedure, it can be shown that the formation of loops is still possible [77]. A node operating the DUAL algorithm participates in at most one update computation per destination at any given time and assumes a stable topology. This is unlike the Jaffe-Moss algorithm that supports multiple concurrent update computations. However, the Jaffe-Moss algorithm can require large bit vectors to track multiple changes in large dynamic networks [78].

2.5.5 Cost comparison of flooding and shortest path routing

Shortest path algorithms consume little transmission resources if networks operate in a static or quasi-static environment. However, as networks become more dynamic, with frequent changes to link costs and topology, shortest path algorithms can be expected to become more costly. This occurs due to high update traffic loads in maintaining accurate routing table information to reflect true link costs and topology. An addition to this update cost is the cost of making routing decisions using inaccurate information stored in routing tables. Not only does this create increased delays, but lost traffic can result in retransmissions from higher layer protocols at source-destination nodes. Shortest path algorithms that maintain routing tables locally within network nodes essentially contribute zero cost in making a routing decision which simply consists of looking up the next node entry in a routing table.

Alternatively, flood search routing techniques contribute zero update cost because they do not maintain any routing tables. As such this class of algorithm does not contribute increased costs as networks behave more dynamically. However, flooding consumes large amounts of network capacity in making routing decisions by searching throughout the network for the destination, every time a user packet is originated. Therefore, large amounts of network resources are consumed resulting in a high routing decision cost.

Table 2.2 compares relative costs expressed as routing table update cost and routing decision cost for general shortest path (SP) and flooding (FLD) algorithms. It is reasonable to conclude that while distributed nonhierarchical shortest path algorithms are less costly to operate in a reasonably static environment, flooding may consume less resources in a very dynamic network. This is especially true if the shortest path algorithm is unable to maintain accurate routing tables due to high rates of change in link cost and topology. By quantifying the relative routing costs between these two classes of routing procedure, it may be possible to identify the point at which the rate of change in a network is such, that it is preferable to use one type of algorithm over another. The objective in the above is to maximise the probability of successfully reaching a destination in a large and possibly very dy-

Table 2.2: Relative Routing Cost

	Static Network		Dynamic Network	
	SP	FLD	SP	FLD
Routing Decision Costs	Low	High	Low	High
Routing Update Costs	Low	Low	High	Low

dynamic network, while minimising the amount of update control traffic or duplicated user traffic.

2.6 Routing Cost and Network Capacity

Constraints imposed by network capacity is the sole reason for routing transmission costs becoming an issue. Any capacity consumed by routing procedures directly affects the user traffic throughput because capacity limits the total available transmission resources within a network. Assuming that user demand exists, user throughput is of prime concern to a network provider as it reflects utilisation of network resources. High utilisation implies efficient use of network infrastructure, essential in networks interconnected by low capacity transmission bearers. From an individual user's perspective, end-to-end transmission delay through the network is a more important parameter as delay reflects the user perceived quality of service [79, p. 299], [80]. While throughput and delay are naturally affected by capacity, they are also greatly affected by the performance of routing procedures. Insight into the relationship between these factors can be gained by examining the delay experienced by user traffic.

Overall average delay experienced by user traffic can be modelled as a network of M/M/1 queues. From Little's Result [81, p. 17]

$$\bar{N} = \lambda T$$

where $\bar{N}$ is the average number of packets in a queueing system, λ is the average arrival rate of packets to the queueing system and T is the average time spent in the queueing system.

Consider the “system” to be a single M/M/1 queue. Arrival rate of packets into the queueing system is described by a Poisson process where the probability of k arrivals in t seconds is given as

$$p(k) = \frac{(\lambda t)^k e^{-\lambda t}}{k!}, \quad k = 1, 2, \dots$$

The average service time to transmit a packet on a link is $1/\mu C$ where $1/\mu$ is the mean packet length and C is the fixed link capacity. Packets are served on a first in first out basis. For simplicity, the service time probability distribution is assumed negative exponential with

$$P\{\text{time to transmit a packet} > t\} = e^{-\mu C t}, \quad t > 0$$

Utilisation or traffic intensity (ρ) can be expressed as arrival rate divided by service rate

$$\rho = \frac{\lambda}{\mu C}$$

The average number of packets in an M/M/1 queueing system can be determined from balance equations, describing flow rates of packets entering and leaving the queueing system, derived from a Markov chain representing the system operating in a steady state [79, p. 128], [81, p. 96].

$$\bar{N} = \frac{\rho}{1 - \rho}, \quad \rho < 1$$

By substituting this expression into Little’s Result, average delay for the queueing system becomes

$$\begin{aligned} T &= \frac{\bar{N}}{\lambda} \\ &= \left(\frac{\rho}{1 - \rho} \right) \left(\frac{1}{\lambda} \right) \\ &= \frac{1}{\mu C - \lambda} \end{aligned} \tag{2.1}$$

Note that $T \rightarrow \infty$ as mean arrival rate (λ) approaches mean service rate (μC), i.e. congestion occurs. Average time spent in a system can also be considered as the sum of time spent waiting in a queue (W) plus transmission time on an outgoing link [82, p. 277].

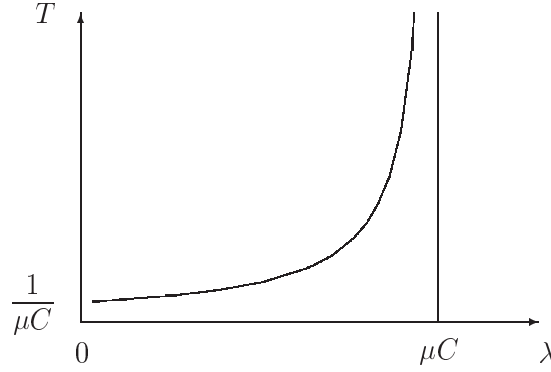


Figure 2.2: Delay versus throughput for an M/M/1 queue

$$T = W + \frac{1}{\mu C}$$

Delay versus throughput for a single M/M/1 queue is shown in Figure 2.2 from [81, p. 98].

Now consider a network of nodes interconnected by links of fixed capacity. Let each outgoing link be modelled as a queueing system, with packets switched using a fixed routing rule. From [83] overall average delay for the network can be derived using Little's Result

$$\begin{aligned} \bar{N} &= \gamma T \\ &= \sum_{l=1}^L \bar{N}_l \\ &= \sum_{l=1}^L \lambda_l T_l \end{aligned}$$

and by rearranging these equations

$$T = \sum_{l=1}^L \frac{\lambda_l}{\gamma} T_l \quad (2.2)$$

where $\bar{N}$ is the average number of packets in the network and $\bar{N}_l$ the average number of packets in service or queued on link l . The average delay in the network is T while T_l denotes average delay on an individual link l and L is the number of links. Total external traffic entering (and leaving) the network is denoted by γ and λ_l is average

traffic flow on any link l . Therefore, average delay T is the sum of all individual link delays T_l weighted by the proportion of traffic flowing on link l . Using Little's Result, the average number of packets waiting for, or using, link l is $\lambda_l T_l$ and the average number of packets in the network is γT .

If it can be shown that traffic flows on different links in a network are independent, then any link l within the network can be modelled as a single M/M/1 queue with arrival rate λ_l and a mean service time of $1/\mu C_l$. The capacity of link l is C_l . However, if packets arrive at a node in sequence on one particular link and then depart on a another link, traffic flows become dependent. This behaviour would naturally occur as traffic passes through the network from one source to the same destination, and as such violate the independence assumption.

Fortunately, when traffic arrives from many sources and is switched through to various outgoing links, the dependence effect is reduced. Packet lengths on each link can be assumed independent and derived from a negative exponential probability distribution. In practice this is true for networks with average degree of connectivity of at least three [82, p. 322]. So by using the result for average delay experienced in a single M/M/1 queue, Equ. (2.1), and substituting into Equ. (2.2), average network delay can be expressed as

$$T = \sum_{l=1}^L \frac{\lambda_l}{\gamma} \left[\frac{1}{\mu C_l - \lambda_l} \right]$$

By further considering routing control traffic in addition to user traffic, propagation delay and node processing time, a more accurate expression for delay can be obtained [82, p. 323]. Let $1/\mu$ denote the average user packet length and $1/\mu'$ denote the average packet length calculated for both user and routing control packets. Traffic flows on individual links, λ_l , now represent the sum of user traffic and routing control traffic flows. Given that

$$T_l = W + \frac{1}{\mu C_l} \tag{2.3}$$

where waiting time, W , is

$$W = \text{average delay} - \text{average service time}$$

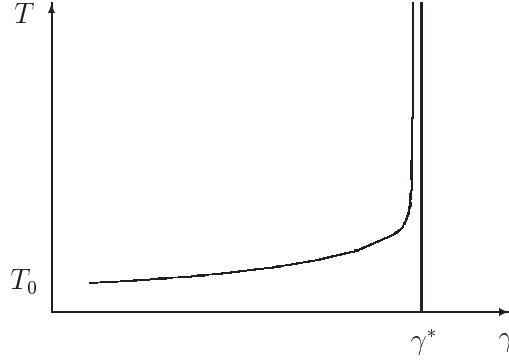


Figure 2.3: Delay versus throughput for network of queues

$$\begin{aligned}
 W &= \frac{1}{\mu' C_l - \lambda_l} - \frac{1}{\mu' C_l} \\
 &= \frac{\lambda_l / \mu' C_l}{\mu' C_l - \lambda_l}
 \end{aligned} \tag{2.4}$$

Substituting Equ. (2.4) back into Equ. (2.3) gives

$$T_l = \frac{\lambda_l / \mu' C_l}{\mu' C_l - \lambda_l} + \frac{1}{\mu C_l}$$

Adding processing delay K and propagation delay P_l , for link l , results in the average delay for any link l being

$$T_l = \frac{\lambda_l / \mu' C_l}{\mu' C_l - \lambda_l} + \frac{1}{\mu C_l} + P_l + K$$

Overall network average delay from Equ. (2.2) becomes

$$T = K + \sum_{l=1}^L \frac{\lambda_l}{\gamma} \left[\frac{\lambda_l / \mu' C_l}{\mu' C_l - \lambda_l} + \frac{1}{\mu C_l} + P_l + K \right] \tag{2.5}$$

The additional K term is required for processing delay in the final node on any source-destination path.

Once the traffic flow in one link approaches capacity, then T_l and hence T will grow rapidly. Overall delay will be dominated by this T_l term in the congested link l and give rise to a threshold type of behaviour shown in Figure 2.3 from [82, p. 323]. Delay remains reasonably constant, approximately the “no load” delay T_0 , until the threshold is reached where T grows rapidly. The saturation load, γ^* , is the critical load at which $T \rightarrow \infty$.

The above discussion assumed a fixed routing rule with saturation load representing one or more links becoming severely congested due to very large queue

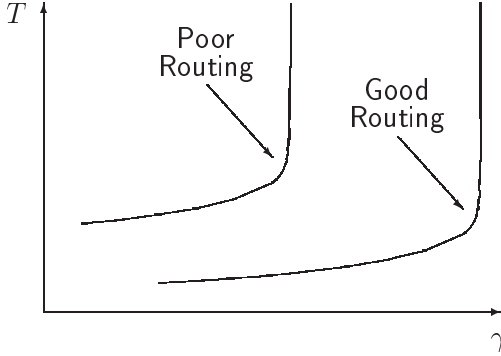


Figure 2.4: Relative performance of routing procedures

lengths. It is highly likely that although some links in the network are congested, others may only be lightly loaded. If routing rules changed or adapted to the local congestion by redirecting traffic via alternate routes, greater throughput could be achieved by effectively increasing the saturation load. This is the motivation behind minimum delay routing which seeks to redirect traffic away from links with large queueing delays.

So, for a given network topology, the routing procedure limits maximum throughput available to the network. If saturation load is approached in a network using a given routing procedure and congestion does occur, flow control mechanisms must be employed to relieve this congestion. Generally, a routing procedure determines the response of a delay-throughput performance curve while flow control operates along this curve preventing the saturation load from being reached, see Figure 2.4 from [79, p. 299].

From Equ. (2.5) saturation load is reached ($T \rightarrow \infty$) whenever $\lambda_l \rightarrow \mu' C_l$ for any link l . Any routing control traffic travelling on link l , as part of a routing table update procedure, will contribute to the total λ_l for that link. As the rate of change to routing tables within a network increase, so will the update traffic flows, therefore saturation load of user traffic entering the network will be reduced. High levels of dynamic behaviour in a network can also lead to uncertainty in routing tables (the convergence time problem) and most likely give rise to retransmissions of user traffic due to lost packets. Therefore, retransmission overheads would further reduce the maximum possible load of useful user traffic.

Threshold behaviour in the delay-throughput curve results from the delay being averaged over a large number of queues in a network, and the absolute value of saturation load is determined by the particular routing procedure being used and total network capacity. Dynamic behaviour in networks, employing distributed adaptive shortest path algorithms, consume network capacity due to update control traffic and retransmissions caused by routing table uncertainty. These overheads (or routing costs) reduce the maximum useful user throughput that can be sustained by a network.

Chapter 3

Combining Routing Procedures

All of the routing procedures described so far tend to perform well only under specific network conditions (such as quasi-static). In practice, some networks can exhibit wide variations in topology change and traffic load conditions from static to very dynamic. It is possible for this behaviour to occur regionally within a network or vary over time. In these cases it is difficult to select a single most efficient routing procedure. One algorithm may prove less costly when operating under more static conditions while another is better suited to dynamic environments. If network behaviour varies from static to dynamic, or the network is dynamic only in particular regions, routing algorithms will be required that can adapt to minimise routing cost. Routing strategies that can adapt to changing network conditions by combining existing routing algorithms into new more effective routing procedures will be explored in this chapter.

3.1 Future Demands on Routing

User demands for wider seamless connectivity and the freedom of mobility result in increased size and complexity of communication networks. The traditional “users” of a network (voice and data terminals) are expanding to include sensors, computer peripherals and servers, while applications require greater integration of voice, video and data traffic within single networks [84, 85, 86]. Increasingly users wish to access information from remote sites or while roaming, demanding terminal and personal mobility services. As telecommunications is relied upon as an essential service, high

assurances of communications availability and survivability are required.

Military users are demanding a seamless integrated information networks for use in strategic, operational and tactical environments. Such a network includes fixed and deployed network infrastructure with distributed radio access systems servicing remote subscribers [87, 88, 89]. Networks must be distributed and highly adaptive to ensure survivability [90]. In addition, routing procedures will be required to cope with intelligently coordinated, non-random topology change patterns to support self-organising network control strategies and improve survivability [91].

These demands will require distributed adaptive network routing methods that not only maintain connectivity between many users in large networks while coping with mobility and network failure, but also reduce operating costs through maximising utilisation and sharing of network resources.

3.2 Background of Combined Routing Procedures

A few routing procedures have been proposed to address the issue of possible variations in the rate of change to offered traffic load conditions. In section 2.5.1 a routing procedure called “Delta Routing” was identified that combined the features of centralised and decentralised routing [13]. In using this scheme, global routing information containing a number of “best” shortest paths to a destination is disseminated to all nodes from a central controller. Nodes are free to make decisions in the selection of a “best” path based on local traffic conditions at the time. This provides an efficient long term strategy based on average performance using the central controller’s routing information while nodes adapt to short term local events caused by individual links and packets. Relative authority (or δ) for nodes to make local decisions is delegated by the central controller. If $\delta \rightarrow 0$, nodes can only use paths specified by the central controller. If $\delta \rightarrow \infty$ then a local shorter queue only scheme is adopted. Choosing an optimum value for δ depends on traffic conditions and network topology, an analytic solution for selecting the value δ had been attempted but not found in [13].

While the dynamic nature of adaptive routing appears to be superior over any

fixed deterministic routing procedure, update traffic overheads and the practical difficulties in assessing traffic conditions make it unclear as to which scheme is more efficient. Deterministic routing procedures are acceptable when traffic conditions can be estimated and remain stationary. Either deterministic or adaptive schemes are satisfactory when stable traffic patterns suffer from short increases in demand from a number of sources, and as expected, adaptive schemes perform better when traffic is unpredictable. A two level approach can improve performance by using fixed deterministic routing rules to determine a number of paths between source-destination pairs and an adaptive scheme to choose a path based on local traffic variations [92]. This strategy must be made insensitive to small variations in traffic conditions.

A similar proposal, also using a two level adaptive routing strategy, utilises a global algorithm to determine a set of “good” paths between source-destination pairs, rather than the “best” path. These paths are selected based on average, slowly varying statistics of traffic and link conditions. Therefore, packets being forwarded along a path at a node will generally have a choice of two or more outgoing links. At the second or lower routing level, a node operates as a multiserver queueing system with each outgoing link serving the queue. Constraints are imposed because some links will inevitably lead to “bad” paths. This scheme offers improved performance over nonadaptive schemes and in heavy traffic conditions improvement by a factor equal to nodal degree was observed [80].

While the above strategies combine different routing procedures to more efficiently operate under varying traffic conditions, network topologies can also be very dynamic. This has led to research in routing strategies that are better able to cope with varying topology change conditions. Two distinctive approaches are evolving in this area, multiple protocol routing and hybrid routing. Multiple protocol routing can be defined as any routing strategy in which one routing algorithm can be deactivated or “switched” in favour of an alternative algorithm or protocol. Protocol switching is based on specified switching criteria and current network state. Hybrid routing describes a routing procedure created from the combination of existing algorithms into a new routing procedure.

In multiple protocol routing, one protocol is chosen for its efficient operation in

a stable network and another more suited to a dynamic network. The two protocols can be regionally employed or switched throughout the entire network at various times depending on the degree of dynamic behaviour in the network. In this way the network can be responsive to a wide range of link cost change behaviour [93]. In static regions (or periods of time) shortest path algorithms may be used to maintain routing tables for forwarding data packets or call requests. This maximises efficient use of network resources because these algorithms are less costly when used in static environments. In very dynamic situations, a flooding protocol may be activated that requires no knowledge of the destination's location and offers maximum survivability. Using controlled flooding when the destination is known to be within a particular region would improve the flooding efficiency.

Finding suitable local parameters to measure the rate of topology change occurring in a network is in itself difficult. In addition, a switchover point must be based on these parameters and the decision function performing switchover must be distributed in a large dynamic network. It can be expected that any parameters used as a basis for the switchover process would be dependent on network topology and the nature of its dynamic behaviour.

An alternative to the multiple protocol routing approach is that of hybrid routing. Hybrid routing combines different routing procedures with their inherent advantages and limitations into a new routing procedure exhibiting more favourable characteristics over varying network conditions. One example of hybrid routing has been proposed for tactical communications to better support user mobility. Each node in the network maintains a global directory of all subscribers and their affiliated nodes. During a call establishment procedure, a call set up packet follows a path through the network towards the destination node, in accordance with routing tables within intermediate transit nodes. If the user being called is not affiliated with that expected destination node, a flood search is initiated through the network. This routing procedure can be shown to offer improved performance over pure flooding and restricted flooding¹ alone [94].

A further comparison of this particular hybrid routing scheme with pure and restricted flooding has been investigated by focussing on network connectivity im-

¹Restricted flooding only occurs in specified regions of a network, such as a quadrant.

plications when the network is subjected to increasingly dynamic conditions [95]. This is in addition to user mobility. It was concluded from these results that pure flooding provided the greatest connectivity capability under dynamic conditions followed by restricted flooding and then the hybrid routing procedure. This result is expected because pure flooding is the most survivable strategy as it attempts every possible path through the network ensuring the destination will be reached. As the hybrid routing procedure was primarily designed to support user mobility, it relied upon shortest path routing to reach the expected destination node. If the path to this destination node becomes disconnected due to topology change and inaccurate routing tables, the call attempt fails. The reliability of flooding in overcoming this limitation by being able to reach any destination was considered by the authors to be a stronger requirement for tactical communications. However, the hybrid scheme did demonstrate throughput gains by a factor of four over pure flooding under static conditions with all nodes and links operational.

Another approach has been proposed for use in situations where rate of change in topology is not so high as to require pure flooding but too fast for shortest path algorithms [96]. In this approach the objective is to build routes quickly so that they may be used before the topology changes and react quickly to establish new routes when topological changes destroy existing routes. A query-response exchange process is used in a table update protocol to seek out new routes by flood search. The flooding is localised and does not generally propagate throughout the network. During execution of the algorithm, loops may form but provided changes cease after a period of time, resultant routes are found loop free and in finite time. The use of flooding in this context is for updating routing tables, not for routing user traffic as has been previously discussed.

3.3 Constituent Protocols for Combined Routing Strategies

Suitable existing network routing protocols must be identified before a combined routing strategy is developed. To be most effective, these constituent algorithms should exhibit complementary attributes within the expected range of varying net-

work conditions. A shortest path algorithm is considered suitable for operation in static and quasi-static environments, although it is anticipated that this type of algorithm will either be unable to keep up with rapid changes in network conditions, or do so with a high routing cost. So, a flood search routing procedure is selected to provide improved characteristics under the more dynamic conditions.

A distributed minimum delay shortest path algorithm exhibits superior survivability characteristics over centralised schemes and by incorporating output queue length metrics into the link cost function, network congestion can be reduced. However, incorrect traffic load information leading to the choice of inferior more congested routes, typically, has less serious consequences than incorrect topological information resulting in the possible choice of non-existent routes [79, p. 342]. Therefore, a minimum hop shortest path algorithm is adopted throughout this thesis where link costs are set to constant values ($1 = \text{link available}$ or $\infty = \text{link failed}$) representing topology connectivity. As the name suggests, a minimum hop algorithm searches for a path with the minimum number of links (or hops) between source-destination pairs. By studying the complex behaviour of multiple interacting routing algorithms using the minimum hop approach, the investigation is simplified while still concentrating on the important topological information update issue.

3.3.1 Distributed minimum hop algorithm

A minimum hop routing procedure can readily be derived from the Jaffe-Moss shortest path algorithm [52]. This algorithm exhibits good convergence performance (see Table 2.1) and is loop free. It is a well known algorithm with its behaviour examined by various authors [37, 54, 78, 76, 77, 96], [4, p. 320]. While in this study the Jaffe-Moss algorithm is being used as a minimum hop algorithm, it is in fact a more general minimum cost algorithm. If the link cost function incorporates parameters such output queue length, then the algorithm calculates minimum delay paths through the network. To execute the algorithm as minimum hop, instead of minimum cost, link costs are constrained to either the value 1 (link available) or ∞ (link failed).

The Jaffe-Moss algorithm operates in a distributed fashion with each node maintaining a distance table and a routing table. The distance table contains costs to

each destination via each outgoing link from the node (i.e. the distance to a destination). Based on this distance table information, the routing table contains the minimum cost and associated outgoing link for each of the destinations. So, for any particular destination the routing table specifies the next node along the shortest path towards that destination. In order to describe the operation of this algorithm the following notation shall be used; this notation is similar to that used in [4, pp. 273–279].

G : a connected graph or network of arbitrary topology

L : number of uni-directional links in G

N : number of nodes in G

v : node executing the algorithm

w : one of node v 's neighbouring nodes

d : a destination node in G

s : a source node in G

(i, j) : a link in G between nodes i and j

$l(v, w)$: current link cost associated with link between nodes v and w

$C_d(v, w)$: cost from node v to destination d , via neighbouring node w

$C_d^*(v)$: minimum cost $C_d(v, w)$ over all w to destination d , for node v

$n_d(v)$: next node along shortest path from node v to node d

Consider a node v in some arbitrary network G executing the minimum hop algorithm; G is illustrated in Figure 3.1 with nodes $\{A, B, C, D, E, v\} \in G$. Node v has neighbouring nodes labelled A, B and E, and the distances in hops to all nodes in the network via A, B and E are displayed in v 's distance table, Table 3.1. One hop corresponds to traversing a single link with unit cost, link costs are shown in Figure 3.1. As an example, the cost or distance measured in hops from node v to destination D via node A is $C_d(v, A) = 4$ and via node B is $C_d(v, B) = 3$. Whereas the cost to destination D via node E is $C_d(v, E) = 2$. So the path via node E offers the least distance to destination D and therefore node E must be the successor node along the shortest path to D. Destination D has a row in both the distance and routing tables of v . The routing table row for destination D indicates that E is the successor node to D and the minimum cost (distance) to D is 2.

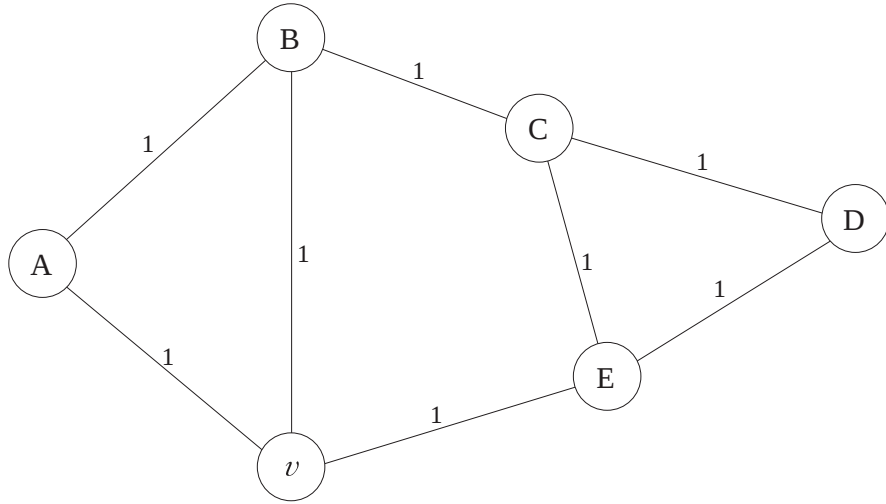


Figure 3.1: Location of node v in the network G

Table 3.1: Distance and Routing Tables for Node v

Distance Table				Routing Table		
d	$C_d(v,A)$	$C_d(v,B)$	$C_d(v,E)$	d	$n_d(v)$	$C_d^*(v)$
A	1	2	4	A	A	1
B	2	1	3	B	B	1
C	3	2	2	C	B	2
D	4	3	2	D	E	2
E	4	3	1	E	E	1

Information contained within routing tables for all nodes collectively specify the set of shortest path spanning trees for all destinations in the network. Note that only local knowledge is contained within the distance and routing tables of this distributed algorithm. As the network's topology changes, link costs change accordingly and nodes update their distance tables to reflect this. If a new minimum cost is found in a node's distance table then the corresponding routing table entry is updated. Update messages are then transmitted to neighbouring nodes informing them of changes to the shortest path spanning tree.

Whenever node v is required to make a routing decision in forwarding user traffic, v simply interrogates the routing table entry corresponding to the destination d as specified in the user packet's header, and transmits the packet on the outgoing link designated as the next node $n_d(v)$. This form of routing is typical of a connectionless

datagram service where user packets contain the full destination address in their packet headers. As link costs change and routing tables are updated, user traffic may take different paths through the network. This can lead to user packets being received by a destination in a different order to the one in which the packets were transmitted by the source. This problem is anticipated when using datagram routing and corrected by higher layer transport protocols on an end-to-end basis.

If this same routing procedure is to be used for virtual circuit routing (connection oriented services) or in a circuit switched network, then routing tables cannot be used to directly route user traffic. This is because connection oriented services demand that packets are received by the destination in the same order as they were transmitted by the source. To overcome this problem and still use the same algorithm to perform route selection, routing tables must only be used for connection establishment. In establishing a virtual circuit between source and destination nodes, connection set up information is exchanged between call control entities distributed throughout the network. Call control utilises routing table information to determine the current shortest path through the network. Switching nodes along this path are then each given a virtual circuit (or call) identifier and the next node to forward user traffic. This information is stored in separate look up tables within each switching node. User traffic is then routed through the network based on a given virtual circuit identifier and its corresponding successor node specified in the look up table. Even though the routing table entries may change, the local look up tables are not altered for the duration of any connection. Call control entities within a network generally communicate via separate signalling channels. This same signalling network can also be used to transport routing table update messages.

From this discussion it would appear that datagram services should in general be routed more efficiently, since as soon as routing tables are updated, user traffic takes the new “shorter” path. Virtual circuit traffic must wait for a new connection set up before any new “shorter” path can be exploited. However, if minimum delay routing is used, oscillations can result in datagram networks because these networks are able to react so fast [79, p. 336]. Traffic in a congested area of the network could be redirected to a lightly loaded area leading to sudden congestion, while the previous area may now be lightly loaded. In virtual circuit routing the average duration of a

connection may be long as compared to table update cycles, so reaction to updates is more gradual, leading to greater stability at the expense of route efficiency.

3.3.2 Jaffe-Moss algorithm implemented as minimum hop

As described earlier, the Jaffe-Moss algorithm consists of two table update procedures. The Independent Update Procedure (IUP), based on a decentralised version of Ford-Fulkerson's algorithm [4, p. 273], is used if an event occurs decreasing a link cost. Jaffe and Moss determined that if link costs decrease, algorithms such as Ford and Fulkerson's are inherently loop free. If however a link cost should increase, then it is possible that loops may form while the algorithm converges. In addition, isolated nodes up-tree of a link failure can suffer from the counting-to-infinity¹ problem discussed on page 18. While the counting-to-infinity¹ problem reduces performance due to increased convergence time and update traffic loads, the formation of loops is also of great concern if it causes excessive delays and increased traffic loads. Loop free routing procedures are also required if routing tables are used for the establishment of virtual connections. In order to prevent the formation of loops, Jaffe and Moss applied a Coordinated Update Procedure (CUP) whenever link costs increase.

To describe the IUP, consider a node v in a network G executing the algorithm. Initially all entries in the distance table are set to $C_d(v, w) = \infty, \forall d, w$ and $C_d^*(v) = \infty, \forall d$. Since no minimum cost is known, $n_d(v) = \emptyset, \forall d$. When a destination d is also one of v 's neighbouring nodes w , entries in the distance and routing tables are set to the following; $C_d(v, w) = l(v, w)$, $C_d^*(v) = l(v, w)$ and $n_d(v) = w$. Whenever node v updates its routing table with a new minimum cost for any destination, it sends a message $\text{MSG}(d, c)$ to all neighbouring nodes, where $c = C_d^*(v)$. If node v itself receives a $\text{MSG}(d, c)$ from a node w , the message is interpreted as c being the minimum cost from node w to d . The following steps are performed by node v in executing the IUP.

¹Infinity is not used in practice, instead some large finite value greater than the longest possible path through the network (e.g. $N - 1$).

INDEPENDENT UPDATE PROCEDURE

- 1) If node v receives a $\text{MSG}(d, c)$ from a neighbour w then
 - (a) Set $C_d(v, w) = c + l(v, w)$
 - (b) Re-evaluate minimum cost to d for all neighbours w . If the minimum cost to d has changed, update $C_d^*(v)$, $n_d(v)$ and send $\text{MSG}(d, c)$ to all neighbours.
- 2) If link cost $l(v, w)$ changes by $\Delta l(v, w)$ then do $\forall d$
 - (a) Set $C_d(v, w) = C_d(v, w) + \Delta l(v, w)$
 - (b) Re-evaluate minimum cost to d for all neighbours w . If the minimum cost to d has changed, update $C_d^*(v)$, $n_d(v)$ and send $\text{MSG}(d, c)$ to all neighbours.
- 3) If link cost $l(v, w)$ changes from ∞ to some finite value then send $\text{MSG}(d, c)$ to node w , $\forall d$.

The algorithm initialises when node v comes up and sends $\text{MSG}(v, 0)$ to all its neighbours. Neighbouring nodes w receiving this message for the first time insert a new row into their distance and routing tables for $d = v$, and a new column for node v as a possible successor node into their distance tables. These neighbouring nodes then update their own routing tables and send messages $\text{MSG}(v, c)$ to their respective neighbours (including back to v). This information eventually propagates throughout the network until all nodes are aware of node v and have calculated next node entries for routing packets towards v . If all N nodes in G repeat this initialisation process, every node will have constructed distance and routing tables with entries for every other node as a destination. The algorithm has then converged providing a set of shortest paths between all source-destination pairs in G .

To operate this algorithm as a minimum hop shortest path algorithm, link costs are set to either unit cost (link accessible) or ∞ (link failed), i.e. $l(v, w) = \{1, \infty\}$. In this regard G can be considered essentially fixed in topology with link costs changing to 1 or ∞ representing changing connectivity status with neighbouring nodes. In a mobile network where nodes are likely to appear as new neighbours, a topology

control entity is required to manage the insertion and removal of columns in the distance tables as required.

In the event of a link cost increase, when the Coordinated Update Procedure is invoked, all nodes uptree¹ of the link on the shortest path tree from a destination d are progressively “frozen.” This begins with the node adjacent to the link and progresses up the shortest path spanning tree. If node v is in a freeze state for a particular destination d , then node v can only update its distance table cost entries for d and must *not* change $n_d(v)$. Node v is not unfrozen until all uptree nodes have increased their respective costs to reflect the link cost increase. Old information is “purged” from the network and since node v cannot choose a node uptree as a $n_d(v)$, loops will not form. This differs from the IUP where an uptree node may have a lower cost due to the fact that “news” of the increase has not as yet propagated uptree.

The mechanism used for the CUP proposed by Jaffe and Moss is to use two variations of update messages, as follows:

$$\text{MSG}(d, c, 0) = \text{no link cost increase, do IUP}$$

$$\text{MSG}(d, c, 1) = \text{link cost increase, do CUP}$$

If $l(v, w)$ increases and $n_d(v) = w$, then node v increases $C_d(v, w)$ as in the IUP. Node v then re-calculates $C_d^*(v)$ and sends $\text{MSG}(d, c, 1)$ to all neighbours and goes into a “freeze” state. Nodes receiving $\text{MSG}(d, c, 1)$ from their current next nodes do the same, gradually freezing all nodes uptree of the link cost increase.

An acknowledgment procedure is used to “unfreeze” the nodes. A node v receiving a $\text{MSG}(d, c, 1)$ from neighbour w does not go into a freeze state if w is not its next node, i.e. $n_d(v) \neq w$.² Instead node v updates its distance table with $C_d(v, w) = c$ and sends an acknowledgment message $\text{ACK}(d)$ back to node w . When a node, such as node w in this case, has received an $\text{ACK}(d)$ from all its neighbours, it now knows that outdated information has been purged uptree from itself. Node w can

¹All nodes with a next node routing table entry for d directed along a path that includes the link with the cost increase.

²node v is then not on the shortest path subtree

then acknowledge its own next node with an $\text{ACK}(d)$ message. At this time w leaves the freeze state and no loops form because all uptree neighbours will have higher costs. The acknowledgment process notifies all nodes that they may unfreeze and subsequently execute the IUP to converge with new shortest paths to d .

In the minimum hop case, the effect is to freeze nodes and set all costs uptree of a failed link to ∞ . Once the freeze cycle reaches the “upper edges” of the tree, new lower cost paths can be identified. So after unfreezing nodes, the IUP can commence and converge to new shortest paths, bypassing the need to count up to infinity by using the coordination. All previously frozen nodes whose next node entries pointed down tree have had their costs immediately set to ∞ . This then enables the IUP to find the new shortest path without looping.

Multiple cost increases may affect a node at once. While one cost is being purged uptree another cost increase may occur for the same destination. Each node must keep track of outstanding acknowledgments expected from each neighbour. To accomplish this, node v maintains a set of bit vectors $\{V_d(1), V_d(2), \dots, V_d(k)\}$ to determine when it is able to leave the freeze state. One bit vector is kept for each of v 's k neighbours, and each destination in G . Every time v receives a freeze message from dntree and forwards the message to all its neighbours, v appends a “0” bit to the right most end of each bit vector. This designates an unacknowledged freeze message has been sent. When v initiates a freeze message due to an increased adjacent link cost, it appends a “1” bit to the right most end of each bit vector. Assume that the various freeze messages sent by v are processed and acknowledged in order by all nodes. Therefore, when v receives an acknowledgment from w , it knows that w has acknowledged the oldest freeze message. To record this event, the left most bit is removed from the bit vector $V_d(w)$ corresponding to neighbouring node w .

Node v uses the bit vectors $V_d(i)$ to decide when to send $\text{ACK}(d)$ to $n_d(v)$. If v receives an $\text{ACK}(d)$ from node w and $|V_d(w)|$ is greater than $|V_d(i)|, \forall i \neq w$,¹ then the freeze message being acknowledged by w has already been acknowledged by all other neighbours of v . If this is the case and the left most bit of $V_d(w)$ is a “0,” v sends an $\text{ACK}(d)$ to $n_d(v)$. If however, the left most bit is a “1” then the freeze cycle

¹ $|V_d(i)|$ is the length of $V_d(i)$ in bits

was initiated by v itself, so no acknowledgment needs to be sent downtree. When using this algorithm for minimum hop routing, a link cost increase only occurs due to link failure, in which case $C_d^*(v) = \infty$ and $n_d(v) = \emptyset$ (assuming it was the adjacent link that failed). As a result, no acknowledgment messages can be forwarded by v if it initiated the freeze cycle due to the failure of an adjacent link. Therefore, there is no need to distinguish between sending a forwarded or initiated freeze message. So for minimum hop routing, the bit vectors can simply be replaced with outstanding acknowledgment counters.

This algorithm is suited to situations when changes in link cost are relatively infrequent but fast recovery is required. If changes occur more frequently, storage requirements can become significant. For a full description of this algorithm and proposed modifications to further improve its performance refer to [52]. The steps required to execute this algorithm for minimum hop routing are as follows:

NORMAL STATE

- 1) If node v receives $\text{MSG}(d, c, 0)$ from neighbour w , or $l(v, w)$ decreases (link comes up), do IUP.
- 2) If node v receives $\text{MSG}(d, c, 1)$ from neighbour w and $n_d(v) \neq w$ then
 - (a) Set $C_d(v, w) = c + l(v, w)$
 - (b) Send $\text{ACK}(d)$ to w .
- 3) If node v receives $\text{MSG}(d, c, 1)$ from neighbour w and $n_d(v) = w$ then
 - (a) Set $C_d^*(v) = C_d(v, w) = c + l(v, w)$
 - (b) Send to all neighbours $\text{MSG}(d, C_d^*(v), 1)$
 - (c) Set $V_d(i) = V_d(i) + 1, \forall i$ with link (v, i) up.
 - (d) Go to FREEZE state.
- 4) If link (v, w) fails, then set $C_d(v, w) = \infty, \forall d$ that $n_d(v) \neq w$.
- 5) If link (v, w) fails, then $\forall d$ that $n_d(v) = w$
 - (a) Set $n_d(v) = \emptyset$

- (b) Set $C_d^*(v) = C_d(v, w) = \infty$
 - (c) Send to all neighbours $\text{MSG}(d, C_d^*(v), 1)$
 - (d) Set $V_d(i) = V_d(i) + 1, \forall i$ with link (v, i) up.
 - (e) Go to FREEZE state.
- 6) If a failed link (v, w) recovers, send a $\text{MSG}(d, C_d^*(v), 0)$ to $w, \forall d$.

FREEZE STATE

- 1) If node v receives a $\text{MSG}(d, c, 0)$ from neighbour w then set $C_d(v, w) = c + l(v, w)$
- 2) If node v receives a $\text{MSG}(d, c, 1)$ from neighbour w and $n_d(v) \neq w$ then
 - (a) Set $C_d(v, w) = c + l(v, w)$
 - (b) Send $\text{ACK}(d)$ to w .
- 3) If node v receives $\text{MSG}(d, c, 1)$ from neighbour w and $n_d(v) = w$ then
 - (a) Set $C_d^*(v) = C_d(v, w) = c + l(v, w)$
 - (b) Send to all neighbours $\text{MSG}(d, C_d^*(v), 1)$
 - (c) Set $V_d(i) = V_d(i) + 1, \forall i$ with link (v, i) up.
- 4) If node v receives $\text{ACK}(d)$ from w then
 - (a) If $V_d(w) > V_d(i), \forall i \neq w$ and $n_d(v) \neq \emptyset$, send $\text{ACK}(d)$ to $n_d(v)$.
 - (b) Set $V_d(w) = V_d(w) - 1$
 - (c) If $V_d(i) = 0, \forall i$, then re-calculate $C_d^*(v)$ and update $n_d(v)$. Send $\text{MSG}(d, C_d^*(v), 0)$ to all neighbours and go to NORMAL state.
- 5) If link (v, w) fails, then $\forall d$ that $n_d(v) \neq w$
 - (a) Set $C_d(v, w) = \infty$
 - (b) If $V_d(w) > V_d(i), \forall i \neq w$ and $n_d(v) \neq \emptyset$, send $\text{ACK}(d)$ to $n_d(v)$.
 - (c) Set $V_d(w) = V_d(w) - 1$

- (d) If $V_d(i) = 0, \forall i$, then re-calculate $C_d^*(v)$ and update $n_d(v)$. Send $\text{MSG}(d, C_d^*(v), 0)$ to all neighbours and go to NORMAL state.
 - (e) If $V_d(i) \neq 0, \forall i$, then set $V_d(w) = 0$
- 6) If link (v, w) fails, then $\forall d$ that $n_d(v) = w$
- (a) Set $n_d(v) = \emptyset$
 - (b) Set $C_d^*(v) = C_d(v, w) = \infty$
 - (c) Send to all neighbours $\text{MSG}(d, C_d^*(v), 1)$
 - (d) Set $V_d(i) = V_d(i) + 1, \forall i$ with link (v, i) up.
 - (e) Set $V_d(w) = 0$
 - (f) If $V_d(i) = 0, \forall i$, then re-calculate $C_d^*(v)$ and update $n_d(v)$. Send $\text{MSG}(d, C_d^*(v), 0)$ to all neighbours and go to NORMAL state.
- 7) If a failed link (v, w) recovers, send a $\text{MSG}(d, C_d^*(v), 0)$ to w for every d .

It is worth noting the following points when executing the above algorithm in the freeze state. Step (5)(e) prevents v waiting for any outstanding $\text{ACK}(d)$'s from w if the link (v, w) to node w has failed. Step (6)(a) prevents node v sending any $\text{ACK}(d)$'s to node w should the link (v, w) eventually come back up again. Finally, step (6)(f) enables a return to the NORMAL state if node w remains the only source of outstanding $\text{ACK}(d)$'s, and this node has been isolated due to link (v, w) failing.

While the CUP prevents loops, it also inhibits the ability of the algorithm to find alternate paths in large networks with frequent cost changes [54]. This inevitably degrades availability due to nodes in the freeze state with $n_d(v) = \emptyset$ being forced to discard packets. Looping on the other hand would simply increase delay and tend to reduce maximum throughput for the network as the frequency of cost changes increases. It is also worth observing that with distributed shortest path algorithms, table update activity occurs independently for different destinations. That is, rows in distance and routing tables can be treated as essentially independent from each other.

3.3.3 Flood search algorithm

The most efficient way to flood an entire network is to use constrained flooding. Constrained flooding uniquely identifies packets associated with a particular flood search by using sequence numbers. Nodes store sequence numbers of packets already flooded. If any packets are received with the same sequence number they are discarded, instead of being further broadcast to neighbours. This technique ensures that all nodes are visited at least once and duplicate packet traffic is kept to a minimum over the whole network. Total traffic flooded through the network from a single initiated flood search is $N(D - 1) + 1$, where N is the number of nodes and D is the network's degree of connectivity [61]. Constrained flooding has been adopted as the flood search routing procedure to complement the Jaffe-Moss minimum hop algorithm under dynamic network conditions because it is the most efficient way to search for any destination without prior knowledge of the network topology.

In the implementation of constrained flooding used for this study, unique sequence numbers are stored in ring buffers within each node. As a new packet arrives, its sequence number is compared with those already in the ring buffer. If the packet received is a new packet, its sequence number will be unique and replaces the oldest number currently stored in the ring buffer. The packet is then duplicated and broadcast on all outgoing links, except the link it arrived on. If the packet arriving is an old packet with its sequence number already present in the ring buffer, the packet is discarded.

Ring buffer length must be sufficient such that the oldest packet associated with a particular flood search in the network has been discarded before its sequence number gets overwritten in the ring buffers. Also, sequence numbers must be long enough that duplication of a sequence number is not likely to occur during the life of a sequence number in any ring buffer. While this can readily be achieved for individual source nodes, there is no guarantee that two source nodes will not generate the same sequence number at the same time. For this reason, nodes need to store source identification together with sequence numbers in ring buffers. Storage requirements for constrained flooding are therefore significant.

3.4 Multiple Protocol Routing

Issues in combining the two constituent routing algorithms introduced in the previous sections, namely Jaffe-Moss minimum hop routing and constrained flooding, into a new routing procedure shall now be explored. The purpose of combining these algorithms into a new routing procedure is to achieve improved performance through the minimisation of routing costs over a wide range of varying dynamic network conditions.

This section examines the concept of multiple protocol routing where routing protocols are switched either globally or regionally throughout the network, depending on the network state. The result is a coarse level of control that generally affects many nodes in the network. An alternative approach discussed in a later section called hybrid routing, operates independently within individual nodes creating a finer level of control that produces a smooth adaptation to a changing network environment.

3.4.1 Operating multiple protocols in one network

Previous discussion has suggested that a distributed minimum hop routing algorithm will cause increases in routing table update cost and routing table uncertainty cost as the frequency of link cost changes within a network increases. Conversely, constrained flooding operates independent of topology change and exhibits a relatively constant routing cost under varying degrees of network change. Flooding overheads are related to the extent of flooding and are contributed by the quantity of duplicated traffic in the network. It is reasonable to consider that for a given network there exists particular conditions when the rate of link cost changes occurring within a network make it preferable to use flooding over minimum hop routing.

In Figure 3.2 the expected relative routing cost between flooding and minimum hop routing as a function of rate of topology change is illustrated. In situations where the network is relatively stable, frequency of link cost changes is low and the minimum hop algorithm operates with low routing cost. This is due to few changes occurring to routing tables within network nodes. As the frequency of link cost changes increases beyond a certain threshold, minimum hop routing may consume

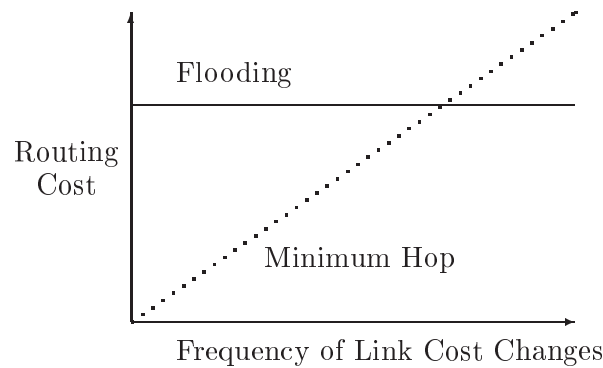


Figure 3.2: Routing overheads as the network becomes more dynamic

more network resources than flooding. Even considering the inefficiency associated with flooding, communications overhead used in tracking topology changes with minimum hop routing could conceivably become excessive in very dynamic networks. Under these circumstances greater routing efficiency leading to higher utilisation may result if routing is performed using flooding. It is these possibilities that motivate the idea of switching routing algorithms as network conditions vary from static to dynamic.

Gains in switching between routing protocols can only be achieved if the global routing cost can be reduced to allow greater network utilisation. So if flooding is selected as being more appropriate under certain conditions, the minimum hop routing table update procedure must be suspended. Otherwise, flooding overheads will simply add to the already high cost of minimum hop routing. Moreover, the cost of flooding must be significantly less than the cost of minimum hop routing to make the strategy worthwhile. In conflict with this is flooding's inherent inefficiency.

The possibility of circumstances existing where flooding can operate with improved efficiency over minimum hop routing is increased if variants of basic flooding, such as controlled flooding, are considered. If controlled flooding is operated with an initial assignment of packet wealth restricting the extent of flooding to only the region of instability, and minimum hop routing is used outside this region, an improved routing strategy may result. For example, if a region in a network becomes so unstable that shortest path routing can simply not maintain accurate routing tables, then switching selected nodes to flooding and assigning a packet wealth sufficient

only for that region effectively operates a “flooding region” within the network. It is essential that unwanted duplicated traffic from the flooding process be kept to an absolute minimum. In order to achieve this, looping packets revisiting nodes can be discarded using a packet sequence numbering scheme similar to constrained flooding.

By integrating controlled and constrained flooding together, assigning a packet wealth of ∞ results in basic constrained flooding of the entire network when the destination’s location is totally unknown. If however, a destination is known to exist in any particular region with some degree of certainty, then a lower packet wealth could be assigned improving the overall efficiency.

This multiple protocol routing concept is illustrated in Figure 3.3 with user traffic forwarded using minimum hop routing tables from a source node s until a node on the fringe of the unstable region in the network is reached. Here nodes (labelled “F”) within the unstable region use constrained flooding to find the destination d . Nodes outside the unstable region that have received flooded packets, forward these packets in accordance with their own routing tables. From the example shown in the figure, this requires nodes to forward packets back along the path that they were received on (assume their routing tables reflect the shortest path). Under these circumstances packets are immediately discarded and the flooding ceases.

If the flooding nodes continue executing routing table update procedures, then routing costs will not be reduced sufficiently to compensate for the added overheads of flooding. However, if these same nodes do suspend routing table maintenance, the impact on other nodes’ routing tables will be significant. Consider a node v in the network that has decided to flood for a particular destination d . Assume v is no longer involved in any routing table update procedures. Neighbouring nodes will not receive any updates from v and as such will not change their distance table entries for v . Some neighbouring nodes may still choose v as their next node to d in which case node v will simply broadcast any received user traffic. As more nodes in the network decide to use flooding for a particular destination there is a much greater chance of incorrect routing tables causing poor routing due to a lack of table updating amongst non-flooding nodes. Moreover, there is a high possibility of deadlocked loop formation amongst nodes left executing the minimum hop algorithm

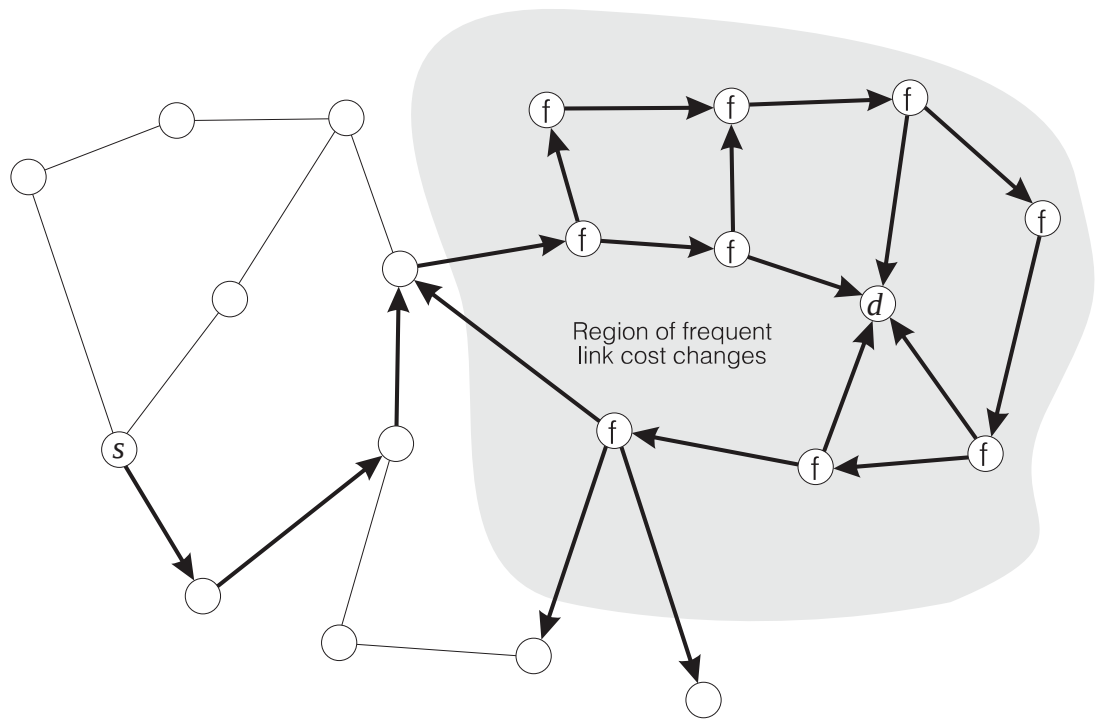


Figure 3.3: Multiple protocol routing

resulting from neighbouring nodes not responding to update messages.

In order to prevent these routing problems from occurring, a more coordinated control strategy is required within the network. Nodes can be identified as belonging to particular regions within the network and forced to execute only one routing algorithm in that region. Boundary nodes are required to provide the necessary routing protocol interfaces between the different routing domains [97]. A higher level control entity is required to monitor the status of network conditions and specify the location of regional boundaries. This approach suggests the need for a centralised architecture to manage the delegation of node roles.

Alternatively, the independence of rows property, characteristic of shortest path algorithms (see page 52), can be exploited by allowing all nodes to flood for only designated destinations. Using this strategy, routing table update procedures are suspended entirely for those selected destinations and continue as normal for all others. The different routing protocols are then assigned on the basis of near static or dynamic destinations.

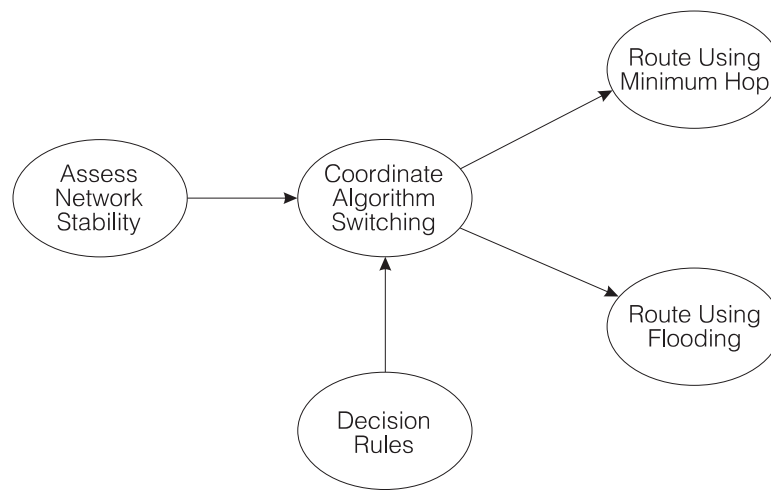


Figure 3.4: Primary node functions required for protocol switching

3.4.2 Problems with alternating between routing protocols

Having discussed the general effects of switching between routing algorithms, let's now consider the issues involved in triggering the switching process. Conceptually, nodes employing a distributed multiple protocol strategy would need to perform the functions illustrated in Figure 3.4. A node monitors network stability using suitable parameters and selects the most appropriate algorithm to use based on a defined algorithm change over rule. The selected algorithm is then activated and used to forward user traffic towards a destination. The parameters used to observe dynamic behaviour in the network must reflect topology, connectivity and traffic load in a form that is local to the node.

Figure 3.2 indicated that rate of link cost changes occurring within a network is an important switching criterion because routing cost is related to the degree of dynamic behaviour in a network. Rate of link cost change can be interpreted locally by monitoring the number of update messages sent and received for a particular destination. Receiving update messages indicates that a neighbouring node changed routing tables, while sending update messages indicates a given node's own routing table has been altered. The latter is more relevant because messages sent from a neighbouring node could relate to another part of the shortest path spanning tree, in another region of the network. Quantities of update traffic generated is also related to the number of outgoing links (a topology parameter), so monitoring routing table

change events that trigger the generation of these update message transmissions is more general.

If a node is routing by flooding, no table update procedures are being executed so another technique is required to assess whether the network has stabilised. Rate of change in adjacent link state is perhaps the only measure. An additional concern is that the isolated nature of flooding precludes any knowledge of performance, so local decisions in one node could very easily cause problems elsewhere in the network severely degrading global performance.

Current user traffic load entering the network provides a measure of added routing cost if flooding is selected. This can be of assistance because high loads could easily saturate the local region. User traffic load entering the network can readily be measured by individual nodes and distinguished from transit traffic passing through a node. It is not possible to measure quantities of retransmitted traffic even though this would provide additional information regarding lost traffic and routing table uncertainty. Retransmissions are generated by higher layer protocols operating end-to-end, of which the network layer has no knowledge. However, coordinated updates used in the Jaffe-Moss algorithm explicitly indicate routing table uncertainty by setting $n_d(v) = \emptyset$, under conditions of no available successor node.

Ability of the minimum hop algorithm to operate effectively when some nodes have suspended routing table maintenance is not possible. Furthermore, true savings in table update costs are negligible when only a small number of nodes are flooding. In these cases flooding increases network load causing further problems for nodes to maintain accurate routing tables. Such issues can be avoided by all nodes changing to flood search routing for particular destinations that exhibit dynamic behaviour. However, this approach is inappropriate in a large network where stable network regions, away from the final destination, could more efficiently make use of routing tables if the neighbouring topology is static.

Many of the parameters used to determine thresholds for the switching of different routing protocols are dependent on network topology and the nature of dynamic network behaviour, including the number of nodes and links, degree of connectivity and topology shape. Moreover, determining absolute threshold values for use in the decision function will most likely only be derived from empirical methods being un-

satisfactory for general network topologies. Observations of simulation experiments in this study highlighted these difficulties in determining effective protocol switching criteria for individual nodes relying on local knowledge. The complex interaction of network elements, dynamic network behaviour, routing algorithm execution, higher layer protocols and a complicated switching process appear unlikely to lead to an efficient robust routing procedure. Implications of observations made during simulation experiments regarding multiple protocol routing are further discussed in Chapter 5.

3.5 The Proposed Hybrid Routing Scheme

Results obtained from investigating the behaviour of the minimum hop algorithm subjected to dynamic network conditions revealed that routing table uncertainty, causing end-to-end retransmissions, actually contributes significantly more to routing overheads than routing table update costs. These findings are discussed in Chapter 5. The implication is that minimisation of routing table update traffic should not be a prime objective. Instead, reducing periods of routing table uncertainty will be more effective.

First generation shortest path algorithms would never directly discard traffic as a result of routing table uncertainty. Traffic would most likely be delayed in loops for a period of time until routing tables converged to reflect current shortest paths. The number of update messages representing table update costs for these algorithms are significantly higher, refer Table 2.1 for Ford-Fulkerson's algorithm. On the other hand, the ability of later algorithms such as the Jaffe-Moss algorithm to rapidly converge to loop free shortest paths is very desirable when used in large quasi-static networks. However, a technique is required to reduce packet discard in the Jaffe-Moss algorithm when routing tables are in a frozen state and no successor node can be found for forwarding user traffic.

Given the assumption that the Jaffe-Moss algorithm causes degraded network performance in very dynamic environments due to routing table uncertainty, and that table update protocol overheads are not as significant; the following strategy is proposed.

Instead of completely switching routing algorithms, as is the case with multiple protocol routing, a hybrid routing procedure is proposed where a local broadcast is initiated if routing tables cannot provide a next node for forwarding user traffic.

This procedure is called hybrid routing because protocols are not switched in the same sense as multiple protocol routing. The shortest path algorithm continues normal execution with a combined local flooding procedure supplementing the deficiencies in tracking dynamic changes to the network.

During normal operation in a quasi-static network, the Jaffe-Moss algorithm provides a next node entry in the routing table for routing decisions. If this entry becomes unknown with no successor node identified due to excessive delays in executing a CUP (i.e. $n_d(v) = \emptyset$), then user traffic is broadcast on all outgoing links (except the link a user packet arrives on). All nodes remain executing routing table update procedures and once the node initiating a CUP has been unfrozen, local broadcasting is no longer used. With this hybrid routing strategy local broadcasts are only invoked by nodes unable to forward user traffic towards a given destination, and local broadcasting is only maintained for the period of routing table uncertainty, ceasing immediately when $n_d(v) \neq \emptyset$. This implies that local broadcast decisions are made on a packet by packet basis within a node, dependent only upon a particular packet's destination address and whether or not a valid successor node currently exists in the routing table for that destination.

Conceptual operation of local broadcast hybrid routing is illustrated in Figure 3.5. Traffic between source node s and destination node d is forwarded using routing table next node entries, except for one node which currently has no successor node designated. This particular node (labelled "?") is required to use a local broadcast to forward duplicate packets towards the destination. Subsequent nodes further downtree towards the destination use minimum hop routing tables because valid next node entries currently exist in their routing tables. The reason a single node in the figure outside the unstable region appears to not forward a packet, is because its routing table successor node is in the same direction the packet was received from. So the packet is discarded.

Sequence numbering of packets, as used in constrained flooding, is required to

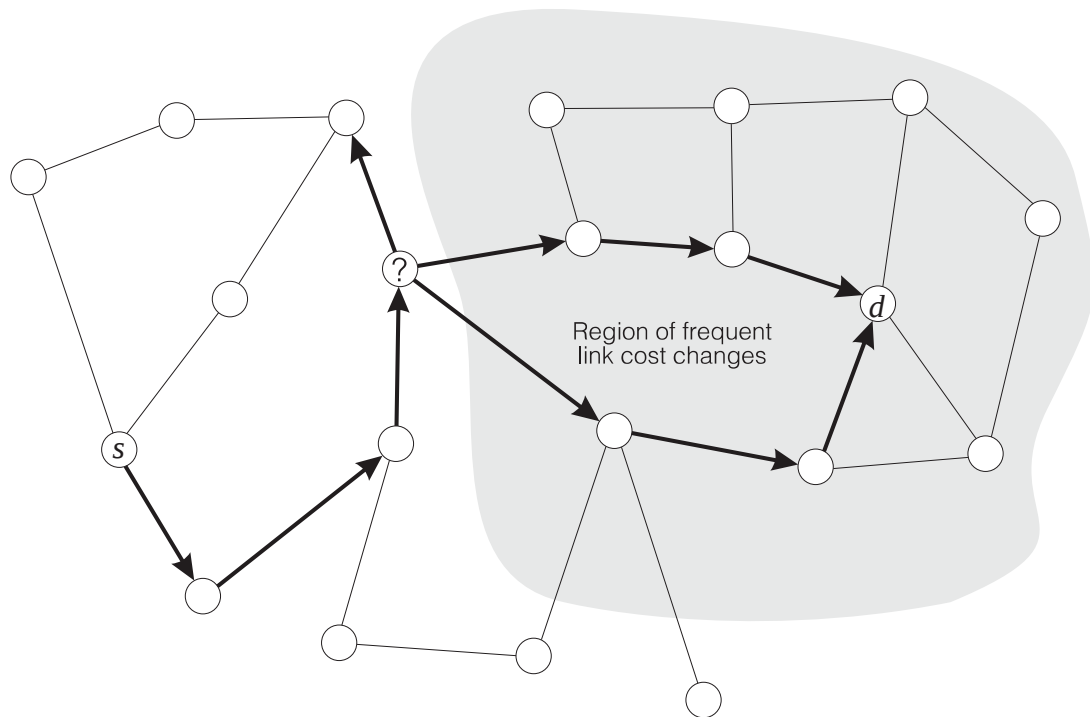


Figure 3.5: Local broadcast hybrid routing

manage the discarding of duplicated packets. This use of sequence numbering also ensures that any packet reaching a final destination does so along a loop free path because any packet with the same sequence number visiting a node more than once is discarded. While packets have improved chances of reaching their destination due to packets travelling along multiple paths, there is no guarantee that at least one packet will reach the destination. This is due to possible looping along all paths resulting in every packet being discarded. However, the probability of reaching a destination is much higher compared to operation of the shortest path algorithm alone.

Hybrid routing operates in isolation when initiating a local broadcast, it assumes no prior knowledge of a destination's location (except that the location is not directed towards a user packet's incoming link). Although a routing table may not contain a successor node, it is very likely that prior information exists within the distance table. Even though there are no assurances of distance table accuracy, distance table information may be sufficient to provide an approximate direction for routing decisions. So while a chosen outgoing link is unlikely to be optimum, at least

it represents the last known direction towards the destination and may lead to more efficient routing than broadcasting. This concept is pursued further in Chapter 6 together with performance improvements to routing table convergence time through priority queueing of control traffic, given that routing table uncertainty has been determined a key issue.

Chapter 4

Modelling of the Routing Strategy

An appropriate modelling technique must be employed to evaluate merits of the proposed hybrid routing procedure. Measuring performance of the new scheme alone is not sufficient because of the difficulty in drawing meaningful conclusions from observations of a single approach. The new scheme must be compared against a baseline under consistent conditions enabling assessment through measured relative performance. This can be readily achieved by comparing the hybrid routing procedure against the constituent algorithms that were used to construct the new procedure. Moreover, as the minimum hop routing procedure is typically used in quasi-static networks and flooding in dynamic networks, it is necessary to compare hybrid routing with these other schemes over wide ranges of dynamic network behaviour. While all these algorithms can be used in general networks, the problems associated with dynamic environments are more pronounced in larger networks. Therefore, only large networks consisting of more than fifty nodes will be considered.

4.1 Methodology

Analysis of adaptive distributed routing procedures is very complex. Not only are there issues of delay performance in time-varying queueing systems to contend with, but also changes in topology and the resultant effect on routing table update procedures loading the network. In addition to this, there is the problem of lost or retransmitted traffic due to routing table uncertainty. While routing procedures may be shown to function well under stochastic dynamic conditions they must also

operate effectively under correlated and other more general patterns of change. Analytical models for adaptive routing have been developed but are constrained to focus more on particular issues, or specific circumstances [98, 99, 100, 101], [16, p. 237]. In general, it is not possible to analyse distributed adaptive routing algorithm performance for typical networks used in practice [82, pp. 318, 424], [16, p. 211]. It is only possible for certain regular graphs such as ring networks [4, p. 314].

A minimum cost distributed shortest path algorithm [35] has been investigated analytically to determine bounds on the number of update messages generated after a resource failure, such as a node or link. The number of update messages have a lower bound given by $\sum_{i,j \in G, i \neq j} (\frac{|C(i,j)|}{2})n(i)$ and an upper bound given by $\sum_{i,j \in G, i \neq j} |C(i,j)|n(i)$, where $|C(i,j)|$ denotes the magnitude of change in a routing table entry during an entire update procedure and $n(i)$ is the number of neighbours to node i [77]. After the recovery of a link, the number of update messages generated is $2(N - 1) + \sum_{i \in G} n(i)|R(i)|$ where N is the number of nodes and $|R(i)|$ is the number of incorrect routing table entries in node i after the resource recovery [76].

Generally, analytical results such as those above are based on synchronous update activity, where updates propagate through the network one cycle at a time with all nodes executing the same cycle of the routing procedure at the same time. Naturally in real systems this does not occur due to queueing, transmission and propagation delays. It can be shown that asynchronous update activity (as in real networks) results in a significantly larger number of update messages than suggested by the above analysis [76]. Moreover, simulation experiments have shown very wide variations in the actual number of update messages generated by these types of distributed algorithms [4, p. 326]. Investigating the complex interaction of asynchronous routing procedures operating under dynamic conditions requires the use of computer simulation modelling techniques. By using simulation it is possible to observe true algorithm behaviour in various dynamic environments and include effects caused by higher layer communication protocols.

Like any form of modelling, simulation has to represent a real system in some form. The processes involved in achieving this can be seen in Figure 4.1, from [16, p. 4]. Complexity of the real system must be reduced by creating a model that abstracts some of the real system's functionality. Since it is not possible to create

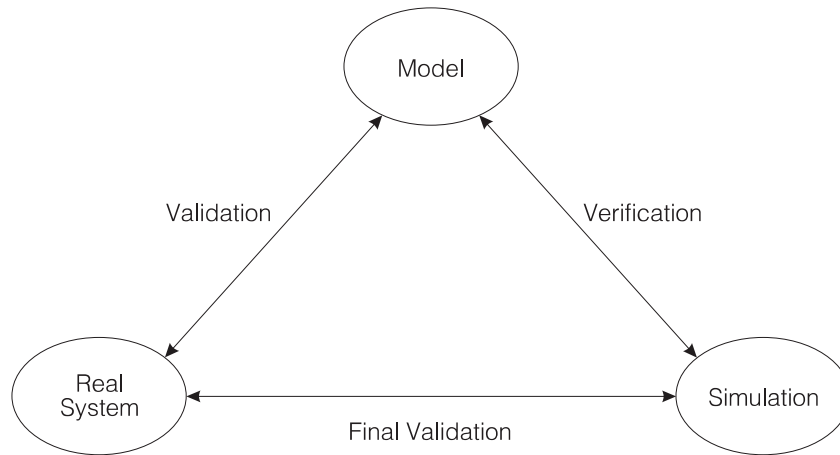


Figure 4.1: Deriving a simulation model

an accurate abstraction under all possible conditions, some assumptions need to be made. The model and its assumptions must then be validated against the real system under the same specific experimental conditions. Once a validated model exists, it is possible to develop a simulation of this model that can be executed so that measurements can be taken reflecting operational performance. The simulation must be verified that it is a true and accurate representation of the model. Final validation of the simulation can be achieved by comparing simulation results with real system performance.

In this study simulation is being used to evaluate the relative performance of specific routing procedures, instead of examining absolute performance of a real communications system. However, a suitable simulation model can be developed by considering the generalised functionality of practical dynamic communication networks. While assumptions can be validated through arguments referring to real network characteristics, it is not possible to conduct a final validation against an existing system. Fortunately, this will not create a problem in the validity of final conclusions because the scope of this investigation is to assess the relative performance of minimum hop, flood search and hybrid routing under identical, reasonably realistic conditions.

While simulation can be useful in investigating the complex behaviour of a system under more realistic conditions, simulation only reflects system performance under

the specific conditions applied to a given experiment. Care must be taken in deriving generalised conclusions from such simulation results. The model developed in the following section is based on stationary stochastic processes, evenly distributed across the network. This avoids, as much as possible, any bias and transient effects thereby reducing the number of experiments required. As a result, final validity of any generalised performance evaluations is improved. Although this is unlike real system behaviour, it enables more accurate analysis of averaged simulation results. Further experiments with more realistic models then follow to explore specific issues of interest.

4.2 Development of the Simulation Model

Development of the simulation model begins by considering possible networking scenarios that could benefit from the proposed routing strategies. Abstractions of key features and functionality can then be used as a basis to derive the actual simulation model.

Mobile packet radio networks, where nodes function as both end user terminals and switches redirecting transit user traffic, generally exhibit dynamic behaviour and require distributed routing procedures. In such networks the nodes are allowed to freely roam and are interconnected by radio transmission links. Networks where nodes are stationary and the users migrate from node to node create a similar routing problem in tracking user mobility.¹ Another dynamic network requiring decentralised routing control can be conceived where fixed and mobile user terminal access is supported by a fixed nonhierarchical communications infrastructure. In this scenario mobile telephone subscribers interconnect directly through radio access points to local telephone exchanges rather than via centralised mobile switching sites as is typically the case today [102, 103].

Typical features of a mobile packet radio network are well suited to the development of a generalised model. Nodes act as sources and sinks of user traffic entering and leaving the network. These same nodes also perform switching and routing con-

¹Deployed military networks as described in Chapter 3, where nodes are stationary while the network is in operation, are an example.

trol functions operating as repeaters for forwarding transit user traffic on towards a destination. The network may be dispersed over a large area with nodes moving around at various speeds. This places demands on radio transmission systems as link quality is dependent on factors such as distance between nodes, terrain, electrical noise and various other forms of interference. Links may slowly degrade becoming unusable, or rapidly fade completely due to terrain blocking a signal. Topology control functions are therefore required to maintain connectivity tables listing current neighbouring nodes. This then enables routing procedures to update local information used for routing user traffic to all destinations in the network. The network could conceivably provide connectionless datagram services or connection oriented virtual circuit type services. The latter requiring a virtual signalling network for connection establishment.

4.2.1 Network and traffic models

The network is modelled as a directed graph G with N nodes and L bidirectional links. Each node functions as a source of user traffic entering the network where traffic can be destined to all other nodes within the network. Routing algorithms are executed within these same nodes and update control messages are exchanged between neighbouring nodes for the purpose of maintaining routing tables. Routing tables are utilised to effectively provide a connectionless datagram service in forwarding user traffic through the network.

As the algorithms being studied are event driven, much of the complexity associated with mobile radio transmission links can be abstracted out through limiting the scope of this study. The routing procedures themselves are triggered solely by link cost change events and it's the relative performance of the routing algorithms that is under investigation, not the overall performance of a real mobile radio network. Therefore, let the network topology of G be fixed with links always existing between nodes and their respective neighbours. Topology change can then be modelled by having link quality change between one of two possible states; a link is either up providing good error free transmission or a link has failed providing no useful transmission at all. This simplified approach requires no topology control as a particular node's neighbours at any given time always belong to the same subset of

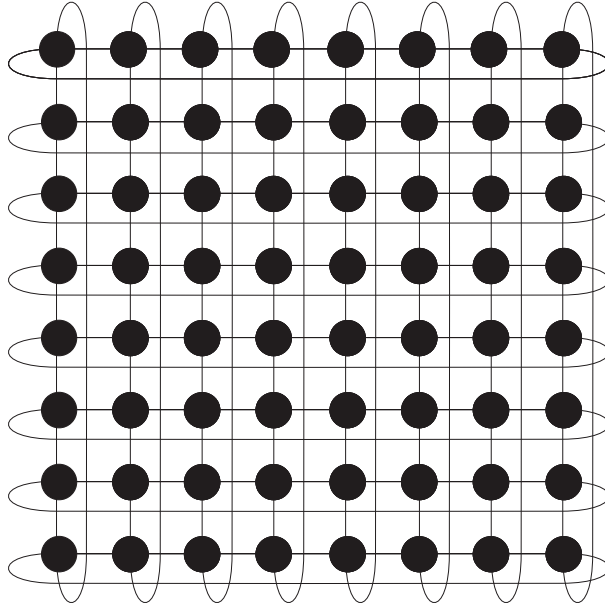


Figure 4.2: Network model - 64 nodes with connectivity of degree 4

nodes as originally defined in G . As a result, effects of topology control do not need to be considered when interpreting results. Routing procedures are still exercised adequately with representative changes in topology achieved through links changing state.

To reduce the number of simulation experiments required, yet thoroughly investigate effects of topology parameters such as network size and connectivity, a limited number of topologies, G , have been investigated. A 64 node network with connectivity of degree 4 has been modelled as shown in Figure 4.2. This network is a reasonably large regular graph forming a manhattan grid network that has been wrapped around itself as a torus to avoid edge effects [104]. To investigate the implications of doubling the network size, a 128 node network, also with connectivity of degree 4 has been studied. Additionally, a 64 node network with connectivity of degree 8 represents a more highly connected graph. In all three cases links are bidirectional.

User traffic entering the network arrives as a Poisson process with arrival rate γ_n at each node n . Each user packet entering the network at node n is randomly assigned a destination node d such that $d \in G$ and $d \neq n$. Therefore, the total rate

of user traffic entering (and leaving) the network is $\gamma = \sum_{n=1}^N \gamma_n$. Total load γ is evenly distributed across all N nodes. User packet length is fixed and set to 128 octets (1024 bits). This is based on the default X.25 Packet Layer Protocol packet size [3, p. 354]. X.25 is typically used in wide area networks and modified versions have been proposed for packet radio networks [105, 106].

It was decided that the user packet size should be significantly larger than the routing table update packet size¹ to effectively capture the overheads caused by user packet duplication when using flooding. This is necessary to more realistically model relative costs between routing table maintenance and flood searches. In datagram networks user packets carry all user data and as such would be expected to be considerably larger than routing table update messages. For virtual circuit routing, the “user” packet represents a call connection request packet and would also be reasonably large containing source and destination addressing, call identifiers and other service negotiation parameters.

As an aside, observations in simulation experiments revealed that smaller user packet sizes do not significantly affect the network saturation load, when measured in bits carried per second. Of course, higher mean packet arrival rates are required to reach this equivalent saturation load. Instead of duplicating fewer large packets when flooding, many more smaller packets are duplicated resulting in similar traffic loads throughout the network.

Transmission links from each node operate error free transmission, assuming that the link is up of course. Outgoing links function as a single server queueing system with service rate defined by packet size and link capacity. Link capacity is fixed at 16kbps to represent a narrowband radio link or perhaps a logical signalling channel. Queues are modelled as having effectively infinite length to ensure that packet discard results solely from routing table uncertainty, possible formation of loops or link failure. A link failure prevents any further transmission in either direction on the link and output queues feeding both ends of a failed link are immediately flushed with all queued packets being discarded. The minimum hop routing algorithm anticipates a possible loss of outstanding update messages and addresses this problem

¹Routing table update messages are set at a fixed length of 8 octets (64 bits) as will be explained later.

in the Coordinated Update Procedure.

The possible discard of user packets creates a problem because the probability of successfully reaching a destination must now be incorporated into the analysis of results. Generally, in real systems lost traffic leads to reattempts or retransmissions by higher layer protocols. These retransmissions will inevitably place additional load on the network and be reflected as an overhead cost, primarily due to routing table uncertainty. To capture this behaviour and provide a more realistic model, a selective repeat Automatic Repeat reQuest (ARQ) protocol has been modelled on an end-to-end basis between source-destination pairs.

When a new user packet is to be transmitted by its source node, a copy of the packet, which must contain a unique packet identifier, is placed in a buffer and a timer is initiated. If a destination node successfully receives this user packet, it immediately returns an acknowledgment packet to the source node. This acknowledgment packet contains the original user packet's packet identifier. Once an acknowledgment has been successfully received, the source node can then stop the timer and remove the copied packet from its buffer. If the original user packet or the returning acknowledgment packet are ever lost in the network, then the timer eventually times out and the copied user packet is retransmitted. A user packet is therefore only removed from the buffer once an acknowledgment is successfully received, this makes multiple retransmission attempts a possibility.

Note that the packet identifiers used for these retransmissions must be different to the sequence numbers used for discarding duplicate traffic as described on page 53. This is because any retransmitted packet using the same sequence number to identify itself, would automatically be discarded every reattempt due to revisiting a previous node. Therefore, a separate packet identifier parameter is required to operate at the higher transport layer while a network layer sequence number is required for managing packet duplication as part of the routing procedures. This use of different sequence numbers at various protocol layers is typical of real data link, network and transport layer protocols and is essential to separate transport and network layer functions to avoid transparency between the protocol layers.

So, there now exists four possible traffic components loading the network. Original user traffic entering the network, returned acknowledgment packets from suc-

cessful transmissions, retransmitted user traffic resulting from lost user packets, and finally, routing table update traffic. While it is possible to distinguish between user traffic and routing table update traffic, it is not possible to differentiate between original user traffic and retransmitted user traffic at the network layer. To do so would provide good local feedback on routing performance but again violate the protocol layering concept and require network nodes to have full knowledge of end user protocols.

4.2.2 Modelling dynamic network behaviour

As has been discussed, dynamic behaviour is modelled by changing link costs in the network G . As links between neighbouring nodes fail and recover, routing procedures are executed and determine new paths as the network topology changes. Instead of roaming nodes, as is the case for real mobile networks, a set of available links essentially move around the network simulating a changing topology. No topology control is required and nodes simply assign distance table columns based on neighbouring nodes as defined in G .

In this model, the amount of change occurring in the network at any given time is described as the intensity of change. Intensity of change, denoted $\mathcal{I}$, is defined as the average number of link cost changes per second, per link. This parameter reflects topology change behaviour that is uniformly distributed throughout the network. Link cost change events are scheduled to occur in the simulation to conform with a given value of change intensity $\mathcal{I}$. To introduce some stochastic behaviour in link cost changes, the interarrival time distribution between link cost changes is assumed to be negative exponential. Mean interarrival time between link cost changes in the overall network is denoted τ where

$$P\{\text{a link cost change occurs} < t\} = 1 - e^{-\frac{t}{\tau}}, \quad t > 0$$

and intensity of change is

$$\mathcal{I} = \frac{1}{\tau L}$$

Note that $\mathcal{I}L$ must be the average number of link cost changes per second occurring in the entire network.

Time τ determines the mean interarrival time between scheduled link cost change events. When a link cost change event is to occur, a single link (i, j) is selected from the L possible links in G using a uniform random distribution. If the current link state of link (i, j) is up, then the link is failed. If the link (i, j) is currently down, it is restored. Both nodes i and j adjacent to the link are notified of this change in link cost. This event then triggers routing procedures in nodes i and j to act upon this cost change for all affected destinations.

A problem arises with this model in that many links can be in the failed state at any given time. This results in network partitioning and the possible isolation of many nodes. While the routing procedures can operate in this environment, gaining insight into algorithm performance is very difficult with severe network partitioning. If on the other hand only one link is allowed to be failed at any time, then partitioning will not occur¹ but the topology will not appear to change very much. What is required is the ability to regulate the average proportion of links that are down at any particular time. This can then be used to control the level of possible network partitioning while allowing sufficient representative change to occur in the topology.

To overcome the above concerns, an additional constraint has been imposed on the network model to limit the degree of change in network topology. Let $\mathcal{D}$ denote the degree of change in topology as characterised in the model. Degree of change, $\mathcal{D}$, is defined as the average proportion of links in G that are in a failed state at any particular time. Therefore, in this model only a fraction of the links, as specified by $\mathcal{D}$, are allowed to be down at any given time. Let τ_f denote the average interarrival time between link failures occurring in the network and τ_d denote the average link down time, then

$$\tau_d = \tau_f \mathcal{D} L$$

and

$$\mathcal{I} = \frac{2}{\tau_f L}$$

The factor of 2 is required here because link failures account for only half of all possible link cost changes that make up the overall change intensity $\mathcal{I}$.

¹as each node is connected to four neighbours

For a given intensity of change $\mathcal{I}$, the mean interarrival time τ_f between link failures can be derived. This value can be then used to characterise a negative exponential probability distribution function. Scheduling a time to select an operational link that is to be failed can be taken from this distribution. Determining the time to schedule this same link's future restoration can also be derived in a similar manner given $\mathcal{D}$ and hence τ_d . A second negative exponential probability distribution can be characterised by τ_d to derive link restoration time. An event list of future link cost changes can be generated using the two probability distributions and provide pseudo random topology changes to the network during a simulation.

Note that a link (i, j) refers to a bidirectional link connecting two neighbouring nodes, so there are two queueing systems associated with each bidirectional link. When a link fails or recovers, the cost change simultaneously affects both directions on a single bidirectional link. In analysing the theoretical M/M/1 queueing model, each bidirectional link (i, j) must be treated as two individual links due to the two separate queueing systems.

4.2.3 Models of routing procedures

Routing algorithms described in Chapter 3 are executed in each of the N nodes in G . Distance and routing tables stored in the nodes are assumed to be configured with all known destinations and neighbouring nodes by some higher level topology control entity. In the actual simulation this occurs during initialisation when nodes are made aware of their own node identity and the identity of all immediate neighbouring nodes. Nodes assume that adjacent link costs are “1” if the link is up and “ ∞ ” if the link is failed. No communication can occur over failed links and nodes assume all output link buffers are emptied whenever a link fails. Both nodes either side of a link are immediately notified in the event that a link cost change has occurred.

The flood search routing procedure is modelled as constrained flooding which is described on page 53. Any user packet transmitted from a source node or received by an intermediate node, that is not the packet's destination, is copied and broadcast on all outgoing links. Intermediate transit nodes do not transmit a broadcast packet back on the same link the user packet was originally received on. There is no advantage in flooding packets back towards the source and to do so only wastes

network capacity. User packets contain the destination node's address in their packet headers.

The operation of constrained flooding requires that any packet associated with a single flood must not be further broadcast if any node is visited more than once by duplicate packets from the same flooding process. To achieve this the routing procedure uses a unique sequence number generated by the source node and stored in the user packet header. Every time a user packet is duplicated for further broadcast, the same sequence number is copied and placed into a new packet header. Nodes store received packet sequence numbers in ring buffers and discard any received packet containing a sequence number already in a ring buffer, without any further flooding. The simulation ensures that each new sequence number generated by a source node is globally unique.

The minimum hop routing procedure model is derived from the Jaffe-Moss distributed shortest path algorithm described on page 42. User packets generated by a source node contain the destination node address in their packet headers. Transit nodes then use routing tables to determine the next node for forwarding packets, based on the destination address in the packet header. If a routing table next node entry is empty, i.e. $n_d(v) = \emptyset$, then the packet must be discarded. Under these circumstances a retransmission timer in the source node will eventually time out resulting in a retransmission of the original packet. Retransmission timeouts have been set at five seconds¹ for all of the simulation experiments performed. In the event of a routing table update when a new minimum cost is sought from the distance table, and more than one node offers the same least cost path, a random selection is made from all nodes offering this same least cost path. This ensures there is no deterministic bias in the minimum hop algorithm.

To notify source nodes that a destination has successfully received a user packet, the destination returns an acknowledgment packet to the source. On receiving an acknowledgment packet, a source node can cancel the retransmission timer for that outstanding user packet. For this to be possible, user packets must also contain the source node address in their packet headers. This is so the destination node

¹The value 5 seconds is an order of magnitude greater than the average "no load" delay and has been chosen to clearly distinguish queueing from retransmission delays.

can place the source address in the acknowledgment packet header for return transmission through the network. As described previously, a transport layer packet identifier is required for source nodes to uniquely identify originated packets being acknowledged. This identifier is a unique number for each user packet transmitted and must be contained in both user and acknowledgment packet headers. The constrained flooding model does not utilise this acknowledgment procedure due to the high probability of successful transmission on first attempt using this scheme.

Routing table update messages used for the Independent and Coordinated Update Procedures are set at a fixed size of 8 octets (64 bits), this being generally sufficient to carry required information such as message type, source address, destination address and routing cost. Valid packet types are message 0 (do IUP), message 1 (do CUP) and acknowledgment. Note that the CUP acknowledgment message is different to transport layer user packet acknowledgments and is used for “unfreezing” nodes. Transport layer retransmission acknowledgment packets are also set to a packet size of 8 octets. This is to reflect the control functionality of this packet type as distinct to packets carrying user information. In a practical system this information may be piggybacked on return user packets.

A model of the local broadcast hybrid routing procedure, see page 60, is largely based on the minimum hop model. If a next node entry is empty, then instead of discarding user packets or retransmission acknowledgment packets, these packets are broadcast on all outgoing links. So in addition to the fields used in packet headers for minimum hop routing, a sequence number field is required because packet duplication may occur. This is managed in the same regard as for constrained flooding.

An alternative scheme, where a next best choice is chosen from distance tables, operates slightly different to the local broadcast scheme. This approach selects the current minimum cost in the distance table and uses the associated node for forwarding user traffic if a routing table next node entry is empty. At first, it appears this scheme should not require a sequence number because no packet duplication can occur. However, using distance tables in this way does not guarantee loop freedom, so to overcome this sequence numbers are used to discard any looping packets revisiting a previous node.

4.3 Comments on Interpretation of Simulation Results

The above model, under the assumptions given, is valid in a general sense to a mobile radio network, except perhaps in the area of modelling dynamic behaviour. The link change model is somewhat artificial with uniform stochastic change occurring across the entire network and is unlikely to be found in a real system. However, a uniform distribution of change with stationary stochastic characteristics enables the derivation of performance metrics through global averaging of statistical results. This assists in drawing more meaningful general conclusions from simulations. Further topology change models representing conditions such as regional topology change and more correlated link cost change patterns will then be required because the performance of routing algorithms is very much related to dynamic behaviour. This will then improve the validity of the model and is explored in Chapter 7.

Computer simulation models have been developed from the above system models. Simulation programs have been developed using the “C” programming language in an event driven simulation framework. A simulation tool called OPNET was used for this purpose. Verification of simulation models is achieved in three main areas. Firstly, extensive incremental testing using packet tracing techniques and monitoring of routing tables was conducted to verify that the simulation models accurately executed the defined routing algorithms. In addition, comparative performance was verified between different routing procedures operating under the same conditions, ensuring consistent delivery of user traffic. A further check ensured that simulation results consistently demonstrate the expected characteristic threshold behaviour as seen in a network of M/M/1 queues. In addition, under static conditions, it is possible to compare and verify simulation results with theoretical results derived from the M/M/1 queueing model. This enables agreement to be reached between a general analytical model and actual results observed from the simulation model.

A global performance measure of a routing procedure’s effectiveness is represented as the maximum user traffic throughput that can be sustained by the network before average delay begins to grow rapidly. Overall expected user quality of service and throughput are important parameters of network performance while the global

averaging of results masks out short term transient behaviour. These results must be compared over a range of change intensities $\mathcal{I}$. Input parameters for the simulation experiments are therefore intensity of change $\mathcal{I}$ and originating user traffic arrival rate per node, γ_n . An output parameter, overall average delay T for the network's user traffic, must be measured. This is done by time stamping all user packets and taking an average of all user packet transit times (including retransmission time) for the duration of a simulation experiment. Average output queue lengths are also monitored to ensure correlation with any increase in average delay.

While γ_n can be a useful indicator of user traffic throughput, it is not an accurate representation. Not all user traffic successfully reaches the destination even though retransmission mechanisms are employed. A probability of successful transmission for node n , p_n , can be derived from the ratio of the number of packets actually received by a destination to the number originally sent by a source node.¹ Let S be the average throughput for any given node in the network and be defined as

$$S = \frac{\sum_{n=1}^N \gamma_n p_n}{N}$$

Throughput is scaled on a per node basis to be consistent with the originating user traffic load parameter, γ_n , and enable comparisons with network models of different sizes.

Network performance is measured as the maximum throughput sustained before the onset of congestion and compared between the different routing algorithms and various intensities of change. Therefore, some point on the delay/throughput curve must be taken as being the threshold at which this occurs. Maximum throughput (and corresponding delay) is taken as the point where a straight line from the origin is tangential to the delay throughput curve. This point represents the limit at which average delay departs from the “no load” delay and begins to grow rapidly, i.e. the “knee” of the curve. Maximum throughput can then be plotted against change intensity for the various routing procedures. However, as intensity of change $\mathcal{I}$ increases, routing table update message load increases as well. This affects the

¹In the simulation model, the measure of the number of packets sent is incremented even if a packet cannot be transmitted by the source. This records the desire to transmit a packet that cannot be achieved by the source (e.g. if $n_d(v) = \emptyset$). Under such circumstances no retransmission timer is activated in the model.

“no load” delay for some routing procedures. So instead of using throughput alone, a metric called routing power [83, 96, 101] is used and is defined as throughput divided by delay (S/T). Any increases in delay caused by update protocol overheads are now incorporated by reducing the routing power. Final comparisons between routing procedure performance over a range of static to dynamic conditions can then be presented in a single graphical plot.

A problem with investigations based on simulation is that statistical results only reflect performance under the particular set of random events that occurred, although for this study relative performance under identical conditions is of prime interest. To increase confidence in simulation results, each simulation run is performed a number of times using different random number seeds with mean value and confidence intervals taken as the final result. Due to the length of simulation run times, it is not reasonable to repeat the experiments too many times. Typically five runs are conducted for each set of experimental conditions. Since less than 30 samples are taken from an assumed approximate normal distribution, confidence intervals for the mean are derived from the t distribution [107, p. 199]. In the operational region of interest, 95% confidence intervals show good correlation in output results for the repeated experiments.

Due to the high processing load associated with simulation experiments causing lengthy run times, simulated duration of network operation was minimised. A simulation duration of 10 minutes was determined experimentally to be sufficient for statistical equilibrium to be reached. This was achieved by running longer test simulations (20 minutes and in some cases 30 minutes duration) for the various scenarios to verify consistency of mean values and confidence intervals with the shorter simulation duration. Although this method of establishing simulation duration can not be treated as statistically sound, it is considered to be sufficient to ensure that for practical purposes of comparing the routing strategies, the inaccuracies in experimental results are minor.

Chapter 5

Performance of Hybrid Routing

This chapter examines the performance of hybrid routing and compares this new routing strategy with constrained flooding and the Jaffe-Moss shortest path algorithm. The latter is used to determine minimum hop paths through the network. These routing procedures and the subsequent development of network models have been previously described in Chapters 3 and 4. Performance of these routing procedures shall be evaluated over a range of network conditions varying from static to very dynamic. It is clear from observed simulation results that as expected the shortest path algorithm is suited to static and quasi-static conditions while flooding is independent of topology changes. Motivation to switch between routing protocols under different conditions becomes apparent from these results. However, the difficulties in distributed decision making make the switching of protocols unattractive. Further discussion of these issues is presented in Section 5.2. Finally, the hybrid routing procedure is evaluated and demonstrates good performance under varying dynamic network environments.

Dynamic behaviour of the network is achieved by randomly causing links to fail and recover across the entire network. A network consisting of 64 nodes with 4 degrees of connectivity is modelled with links operating at 16kbps. The network is therefore interconnected by 128 bidirectional links. For any particular intensity of change $\mathcal{I}$, there are on average 5% of links failed at any given time, i.e. $\mathcal{D} = 0.05$. This represents reasonable change to the topology without causing significant network partitioning. Other values of $\mathcal{D}$ are examined in Chapter 7.

User traffic packet size is fixed at 1024 bits with routing table update messages

and user acknowledgment packets having packet sizes of 64 bits. User traffic entering the network is subject to a Poisson arrival process of mean rate γ . This traffic load is evenly distributed across all N nodes with external traffic entering an individual node n at mean rate γ_n , $\gamma_n = \gamma/N$. Simulation results are plotted as a function of this scaled originating user traffic arrival rate per node. Total network throughput is similarly scaled by a factor of N representing average throughput per node, S . By scaling parameters on a per node basis, results reflect individual users' (nodes') average delay/throughput quality of service. An end-to-end selective repeat ARQ protocol using a retransmission timeout value of 5 seconds is applied to all user traffic transmissions.

The duration of simulated network time is fixed at 10 minutes for the collection of output statistics. This duration is sufficient to achieve a steady state in all output results. Actual simulation experiment run time was considerably longer than this and many experiments were required to cover the range of input parameters with various random number generator seeds.

5.1 Performance of Minimum Hop and Flooding Protocols

Before evaluating the hybrid routing strategy, performance baselines are required and these have been determined using the two algorithms from which the hybrid scheme has been derived. In addition, since the minimum hop and flooding algorithms are being used to establish a performance baseline, these algorithm models must be verified to ensure that simulation results are accurately representing the network under investigation. Simulation models are initially verified by comparing results with the queueing theory analytical model under static conditions. Other verification tests conducted were discussed in Chapter 4.

5.1.1 Static network conditions

Simulation experiments of flooding and minimum hop algorithms using the 64 node network model under load yielded the results shown in Figure 5.1 for a static net-

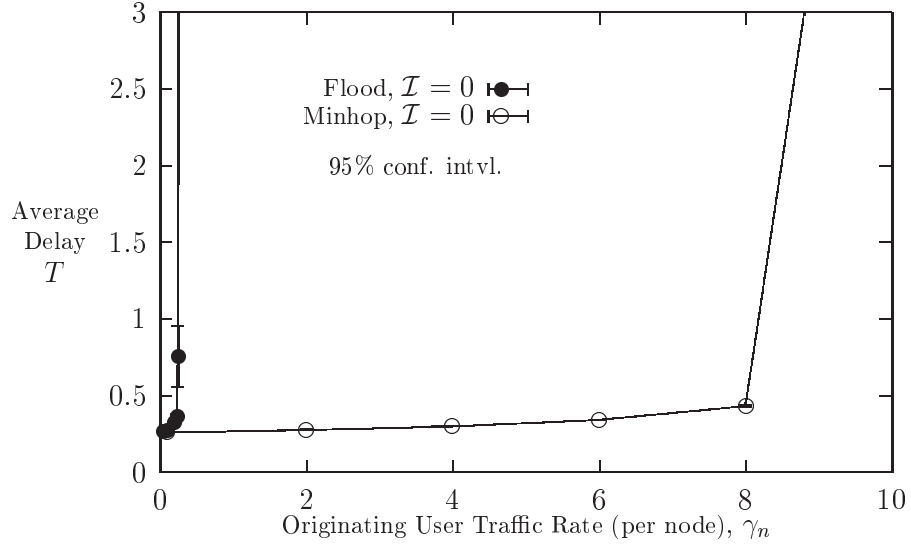


Figure 5.1: Delay using flooding and minimum hop in static network

work. Network topology remained constant ($\mathcal{I} = 0$) throughout the duration of these experiments. Duplicate packet overheads in constrained flooding are the cause of reduced traffic loads able to enter the network with reasonable average delay performance. The minimum hop algorithm requires no network resources for routing table updates¹ as $\mathcal{I} = 0$ and the network is static and therefore operates with a far lower routing cost allowing significantly higher user traffic loads.

The characteristic threshold behaviours of both curves in Figure 5.1 are similar to those exhibited by a network of M/M/1 queues as shown in Figure 2.3. Although simulation experiments use fixed length packets corresponding to an M/G/1 queueing system, a network of M/G/1 queues exhibits the same threshold behaviour demonstrated by the M/M/1 queueing model [82, p. 324].

Onset of saturation load with the minimum hop algorithm developed as some queues in the network became severely congested. This routing procedure is focussing on network connectivity and not adapting to traffic load. It is highly likely that even with the uniform distribution of user traffic entering the network, some links will only be lightly loaded. Given these circumstances, minimum delay routing could be expected to offer improved performance by redirecting traffic onto the

¹except for the initialisation of routing tables

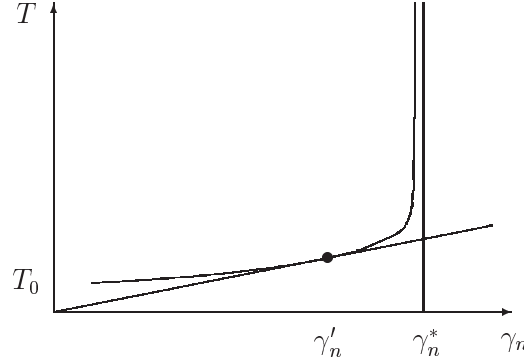


Figure 5.2: Determining the maximum operating point, γ'_n

lightly loaded links further increasing possible saturation load. By using the Jaffe-Moss algorithm as a least cost routing procedure and defining a link cost function consisting of connectivity and queue length parameters, overall performance could be improved. However, as discussed in Chapter 3 this introduces more issues and variables further complicating the study while connectivity alone is of prime concern to dynamic mobile networks. Therefore, minimum delay routing adapting to traffic overload conditions has not been pursued in this work.

Using saturation load γ_n^* to assess an algorithm's performance is somewhat unrealistic because a network cannot be operated with infinite average delay, refer Figure 5.2. A more realistic measure is the maximum operating point, denoted γ'_n , which is the maximum user traffic load entering the network before average delay departs from the “no load” delay and begins to grow rapidly. This is in effect just before the “knee” of the delay-throughput curve. Maximum operating point γ'_n can be defined as the point where a line from the origin is tangent to the delay-throughput curve [83, p. 9] and is shown in Figure 5.2. Note that γ'_n is not true throughput but the maximum operating originating user traffic arrival rate. While true throughput should strictly be considered when comparing routing procedure performance, the maximum operating point expressed as originating user traffic load is essentially the same. Appendix A shows that the probability of successful transmission at γ'_n is close to 1. Therefore, throughput can be approximated by the input parameter γ'_n .

Returning to Figure 5.1, throughput γ'_n is 0.2 packets per second for flooding and in the case of the minimum hop routing procedure $\gamma'_n = 8$ packets per second.

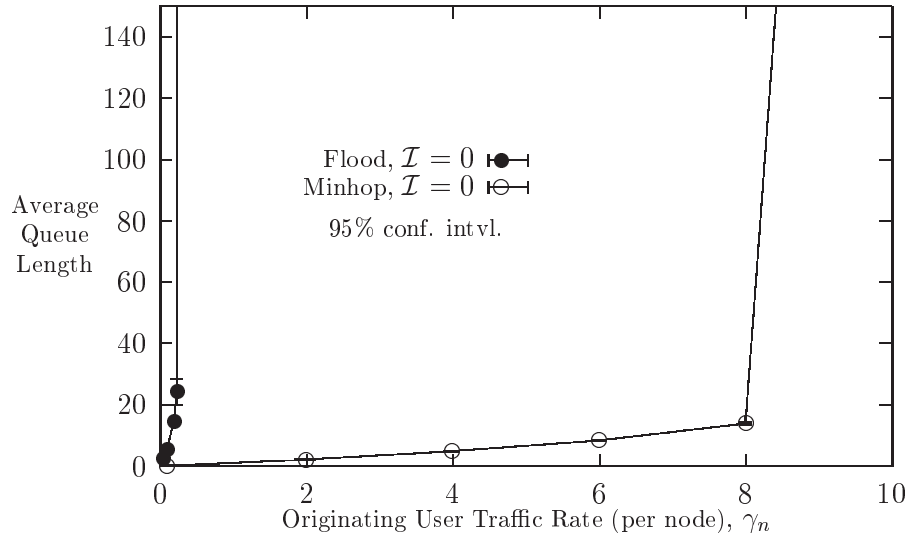


Figure 5.3: Queue length with flooding and minimum hop in static network

Flooding severely constrains the possible useful throughput that can be sustained by the network. This is due to the high routing cost associated with flooding as it searches the entire network every time a routing decision is required. In a static network operating shortest path routing (to determine routing tables), performance is vastly superior when compared with flooding. The minimum hop scheme simply refers to routing tables to make routing decisions and these tables are fixed in a static network. The routing table update cost is zero under these circumstances.

Threshold behaviour exhibited by both flooding and minimum hop routing procedures has been described as being caused by queueing delays. This forms the basis of the analytical model. Therefore, it is expected that average queue lengths in the network model should increase with corresponding increases in average delay performance. To verify this, average queue length statistics have been collected from the simulation experiments and these results correlate with the average delay performance. This can be demonstrated by comparing Figures 5.1 and 5.3 which clearly show the strong correlation between average delay and average queue length performance. Average queue length in Figure 5.3 is measured in units of 64 bit data blocks.

5.1.2 Verification of the simulation model under static conditions

Threshold behaviour of delay/throughput characteristics in a network of M/M/1 queues occurs when one or more links become severely congested. Since delay performance is averaged over all queues in the network, average delay only grows rapidly under highly congested conditions. This is why average delay increases dramatically beyond some threshold network load. This effect was illustrated in Chapter 2, Figure 2.3 and derived from the following expression

$$T = \sum_{l=1}^L \frac{\lambda_l}{\gamma} \left[\frac{1}{\mu C_l - \lambda_l} \right] \quad (5.1)$$

Determining λ_l flow rates in dynamic networks with changing topologies, additional control traffic and retransmissions, is very difficult. This is the prime reason why simulation was chosen as a tool for investigating the issues in this thesis. However, under static conditions values for λ_l can readily be determined and used to verify the simulation model.

Individual link delays T_l and hence T , tend to infinity as $\lambda_l \rightarrow \mu C_l$ for any link l . Links are assumed to have zero propagation delay and nodes zero processing delay. These assumptions are used in the simulations. Mean service rate, μC_l , represents transmission delay on link l . For the M/M/1 model, packet lengths conform to a negative exponential probability distribution of mean size $1/\mu$.

Consider now verification of the minimum hop routing procedure simulation model. If γ is the total traffic flow entering (and leaving) the network and γ_{jk} represents traffic flow from nodes j to k along a particular path, then

$$\lambda_l = \sum_{j=1}^N \sum_{k=1}^N \gamma_{jk} ; j \neq k, l \in \pi_{jk} \quad (5.2)$$

with source-destination nodes j and k such that the path between nodes j and k (π_{jk}) contains link l .

Furthermore, if total traffic entering the network is evenly distributed between all nodes, then any particular source-destination traffic must be the total network traffic flow entering the network divided by the number of nodes and number of destinations.

$$\gamma_{jk} = \frac{\gamma}{N(N-1)}$$

If each average source-destination flow γ_{jk} is now exactly the same, then

$$\lambda_l = f_l \frac{\gamma}{N(N-1)}$$

where f_l is the number of γ_{jk} flows on link l . For a static network where the routing tables are fixed it is possible to determine the exact number of source-destination flows on each individual link in the network. Using this set of f_l values, average delay for the network can be calculated for various traffic loads γ .

To give the simulation model identical characteristics to this analytical model, slight modifications were required. Instead of using a fixed user packet size of 1024 bits, packet lengths were derived from a negative exponential probability distribution with a mean packet length of $1/\mu = 1024$ bits. User packet acknowledgments and retransmissions were disabled for comparison with the M/M/1 model.

Delay versus user traffic load for both the M/M/1 theoretical model and the simulation experimental results are shown in Figure 5.4. Close agreement is evident between the two curves verifying that the simulation model is behaving as expected under static conditions.

As explained in Chapter 4, it is not possible to determine the λ_l values under dynamic conditions with routing tables changing after link cost change events. Determining update and retransmission traffic flows further complicates the problem. However, basic operation of the simulation model has been verified.

In the case of constrained flooding, it is not possible to directly apply the M/M/1 model as defined by Eqs. (5.1) and (5.2) because packets do not follow a single path through the network. Assume a general flooding procedure results in a broadcast packet being transmitted exactly once on all L links. Therefore, every originated source packet generates one packet on every link in the network. This routing procedure effectively changes the constraints on Equ. (5.2) giving

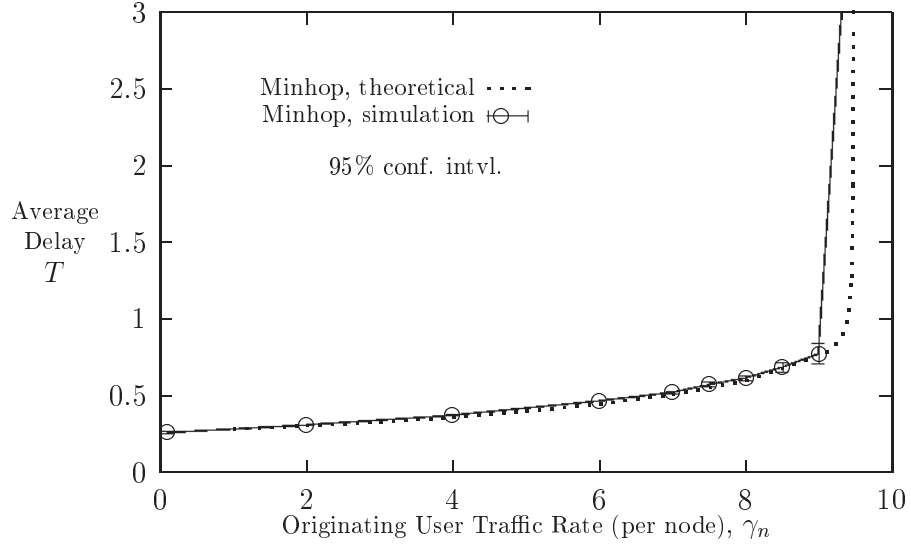


Figure 5.4: Verification of minimum hop in static network

$$\begin{aligned}
\lambda_l &= \sum_{j=1}^N \sum_{k=1}^N \gamma_{jk}, j \neq k \\
&= \gamma_n N \\
&= \gamma
\end{aligned}$$

With constrained flooding, packets are discarded if they revisit a previous node. Therefore, λ_l can be reduced proportionally. The actual number of packets generated per an originated flood search is $N(D - 1) + 1$ where D is the average degree of connectivity, see Page 53. So, for constrained flooding let

$$\lambda_l = \frac{N(D - 1) + 1}{L} \gamma$$

Substituting λ_l back into Equ. (5.1) will not yield overall average delay as perceived by the users. Instead, the average delay experienced by all duplicated flooding packets is determined. The user only perceives the delay experienced by the first packets that actually reach a destination.

While Equ. (5.1) accurately models the threshold behaviour caused by the total number of packets in the network, it will not reflect the true “no load” delay perceived by end users. True “no load” delay T_0 occurs when the network is lightly

loaded and on average there is no waiting in queues. Therefore, T_0 can essentially be expressed as

$$T_0 = \frac{\bar{h}}{\mu C_l}$$

where $\bar{h}$ is the average number of hops through the network.

It is proposed that for constrained flooding, average delay determined by Equ. (5.1) be scaled to more accurately reflect user quality of service while still preserving the effect of many queues becoming congested. Overall average delay expressed by Equ. (5.1) is scaled by the total number of links producing a per link average delay. This per link average delay can then be multiplied by the average number of hops to reflect the average user's perceived delay. Therefore, for constrained flooding overall average delay is expressed as

$$T = \frac{\bar{h}}{L} \sum_{l=1}^L \frac{\lambda_l}{\gamma} \left[\frac{1}{\mu C_l - \lambda_l} \right] \quad (5.3)$$

Average number of hops, or average path length $\bar{h}$, can be derived from the sum of all shortest path distances to every destination (from any source node) divided by the total number of destinations. Shortest path distances to all destinations are contained within the minimum hop routing tables in each node. Therefore, the average number of hops in this network can be found as

$$\bar{h} = \frac{\sum_{d=1, d \neq v}^N C_d^*(v)}{N - 1} \quad (5.4)$$

For the network model currently being simulated, $\bar{h}$ was calculated from Equ. (5.4) to be 4.06 hops. This average path length can now be substituted into Equ. (5.3). Note that the total number of links L used in this model must be twice the number of bidirectional links to accurately represent the total number of individual queueing systems.

Theoretical and simulation model results of delay versus user traffic load performance for constrained flooding are shown in Figure 5.5. Close correlation of maximum operating points between the two curves supports simulation model verification.

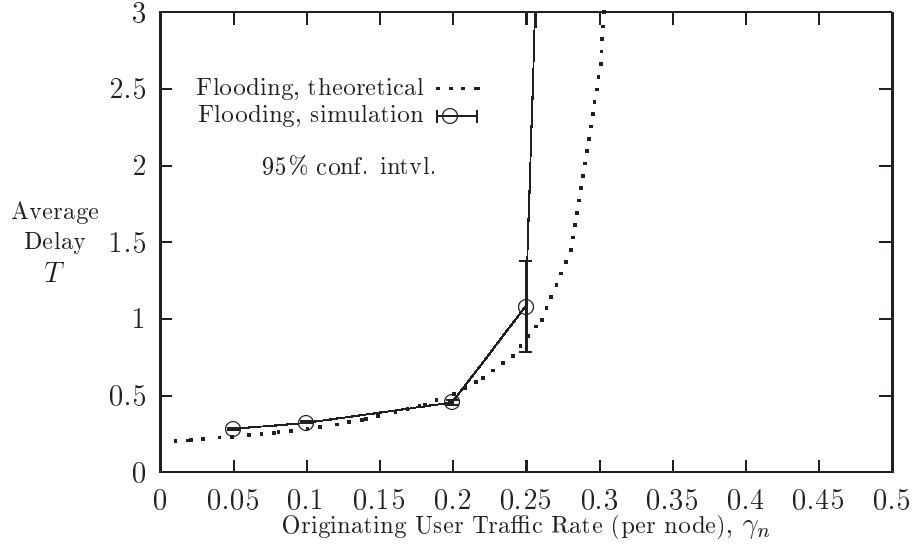


Figure 5.5: Verification of constrained flooding in static network

5.1.3 Effects of dynamic network conditions

Consider now the performance of flooding and minimum hop routing procedures operating in a dynamic network environment. This is achieved in the network model by increasing the intensity of change parameter $\mathcal{I}$. Recall intensity of change $\mathcal{I}$ refers to the average number of link cost changes per second, per link. Within the context of the network model used here (as opposed to the analytical model above), the term link refers to a bidirectional link connecting two neighbouring nodes and there are two queueing systems associated with each bidirectional link. When a link fails or recovers (i.e. a link cost change occurs) the cost change simultaneously affects both directions (queueing systems) on the single bidirectional link.

Response of constrained flooding to increasing intensity of change is illustrated in Figure 5.6. As $\mathcal{I}$ increases the maximum operating point remains essentially the same with $\gamma'_n = 0.2$ packets per second. This implies that performance of constrained flooding is independent of changes to network topology, as expected, because flooding requires no knowledge of network topology to operate. For these reasons, flood search routing is widely adopted for use in dynamic military networks [95]. It appears that although the maximum operating point remains the same, the threshold behaviour is not as sharp under very dynamic conditions (e.g. when

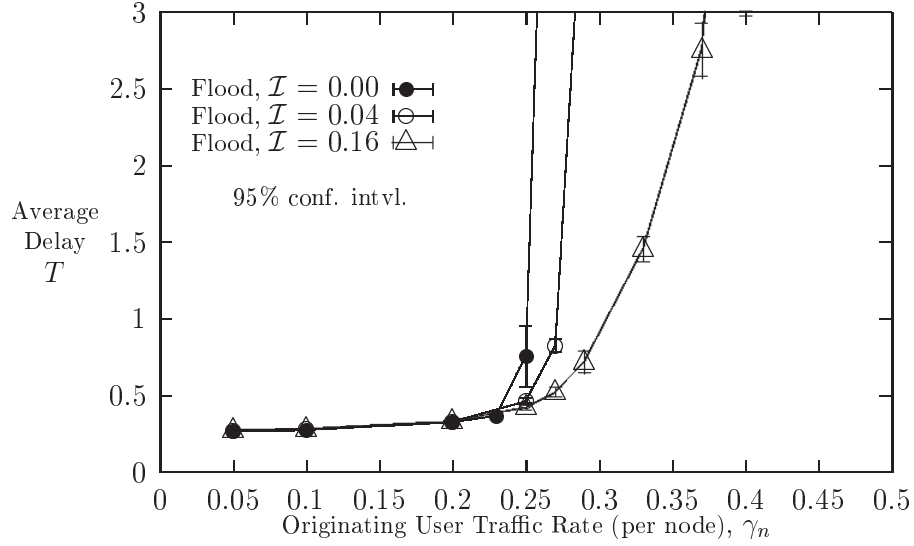


Figure 5.6: Delay using flooding with increasing intensity of change

$\mathcal{I} = 0.16$). This is due to frequent queue flushing as links fail more often, artificially relieving queue congestion.

As expected, performance degrades in the case of minimum hop routing with a network subject to increasing intensity of change. This can be seen in Figure 5.7 for $\mathcal{I} = 0.0, 0.02$ and 0.04 . As intensity of change increases more network resources are consumed in maintaining routing tables while the routing algorithm tracks changes in topology. The routing update cost increases with higher values of the intensity of change. These higher update costs reduce available network capacity for user traffic thereby reducing the maximum operating point γ'_n . A threshold like behaviour is still exhibited by the routing procedure under these circumstances.

If intensity of change is further increased, the minimum hop algorithm no longer exhibits the same characteristic behaviour described by the queueing theoretical model. This departure from threshold type behaviour is shown in Figure 5.8 when intensity of change is set at $\mathcal{I} = 0.06$. Increase in average delay and the high variance in output statistic samples is due to effects caused by packet retransmission. On average many user packets are requiring at least one retransmission in order to successfully reach their respective destinations. The retransmission timeout delay is significantly affecting the average delay performance. Signs of this behaviour are

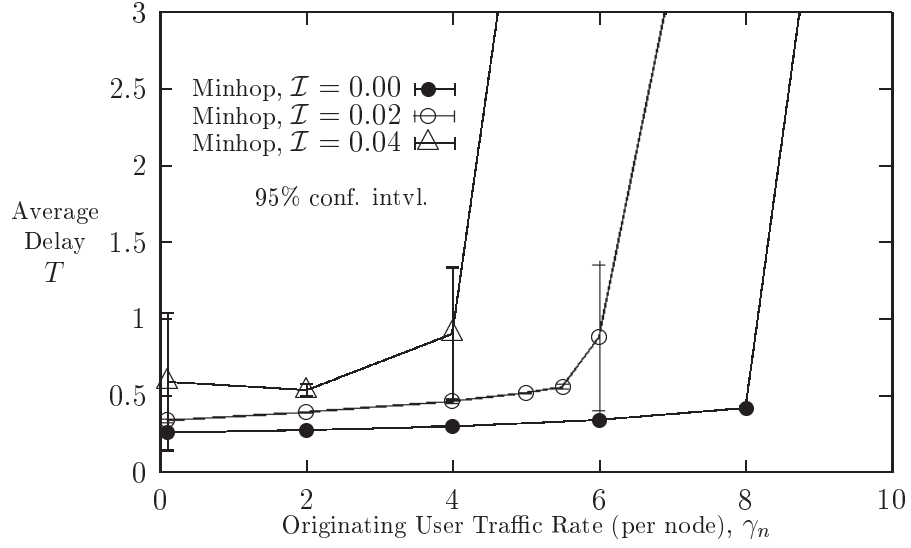


Figure 5.7: Delay using minimum hop with increasing intensity of change

becoming evident in Figure 5.7 for $\mathcal{I} = 0.04$.

Since retransmission timeout parameters begin to dominate the average delay performance, overall performance is in effect being dictated by higher layer transport protocol performance. Essentially the routing procedure is ineffective in delivering reasonable throughput under these dynamic conditions. For these higher values of change intensity the minimum hop routing procedure is inappropriate.

As mentioned previously γ'_n does not reflect true throughput but instead originating user traffic load entering the network. Maximum average throughput S (scaled on a per node basis) is defined as the actual amount of traffic successfully delivered to destination nodes when the network is operating at its maximum operating point, for a given intensity of change. This was described in Chapter 4 but is now more accurately expressed in the following equation

$$S = \frac{\sum_{n=1}^N \gamma'_n p_n}{N} \quad (5.5)$$

Here the originating user traffic load at the maximum operating point is weighted by the corresponding proportion of traffic successfully received and scaled by the number of nodes. Probability of successful transmission, p_n , can readily be measured during the simulation experiment by monitoring the proportion of packets received

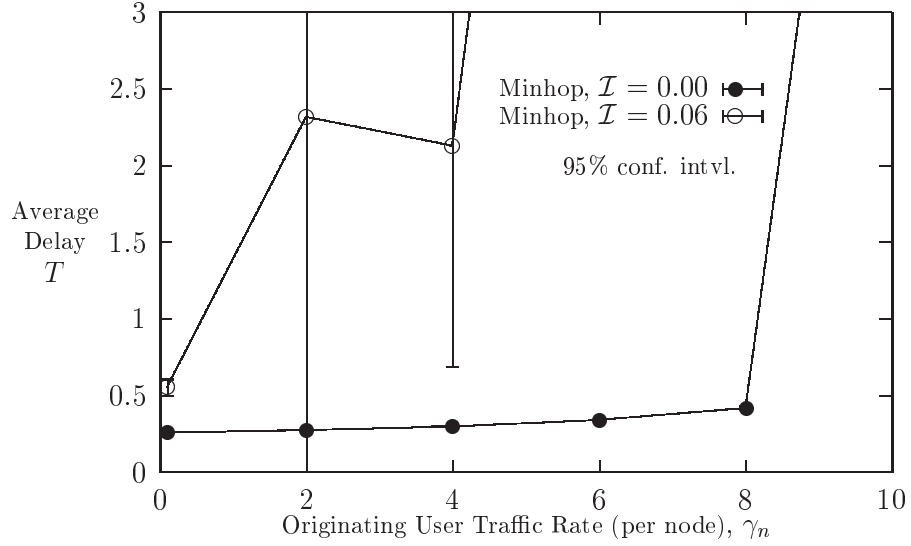


Figure 5.8: Dominance of retransmissions with increasing intensity of change

to those attempted to be sent. A successful transmission is registered whenever a source node receives a user acknowledgment packet corresponding with a previously originated user packet.

An output parameter measuring the probability of success from simulation experiments is used to derive the network throughput for comparison of routing strategies using Equ. (5.5). Probability of successful transmission under increasing load and intensity of change is shown in Figure 5.9 for the case of minimum hop routing. Maximum operating point γ'_n is obtained from the delay performance curve and used to derive the corresponding probability of success. These parameters can then be used to calculate the actual throughput for a given routing procedure and intensity of change. Note that Figure 5.9 correlates well with Figure 5.7 as should be expected.

In summarising the results obtained in this section, flooding is independent to changes in network topology but very inefficient when compared to shortest path routing. However, shortest path routing can only be operated in static or quasi-static conditions where topological changes occur less frequently. Maximum operating point reduces as intensity of change increases with shortest path routing due to routing table update traffic and retransmission traffic load. As intensity of change

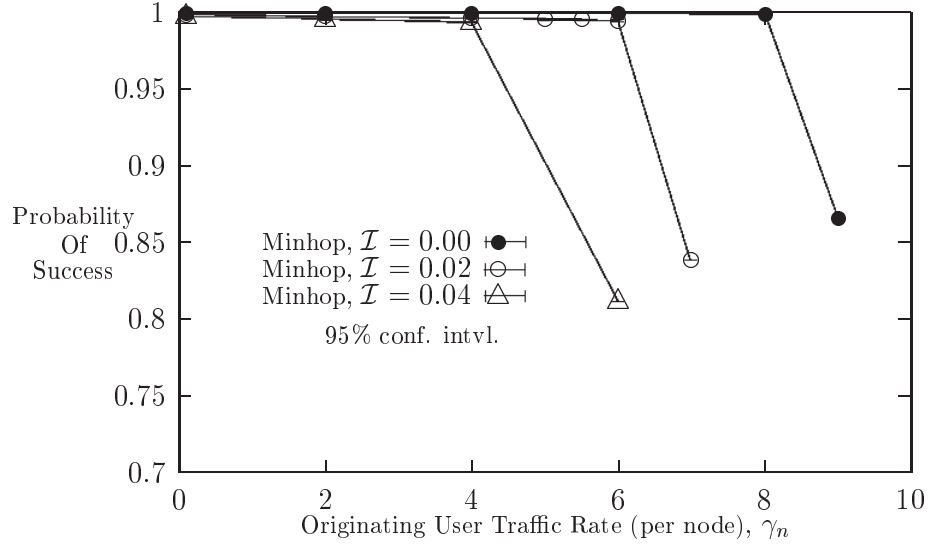


Figure 5.9: Probability of success using minimum hop with increasing intensity of change

increases further, retransmission traffic begins to dominate shortest path routing procedure performance. It is reasonable to conclude that a large network similar to the one modelled, would require a flooding procedure if the network is to operate in a very dynamic, or potentially very dynamic environment. Unfortunately, flooding results in very low network utilisation, and therefore better routing strategies need to be sought.

5.2 Implications of the Results to Multiple Protocol Routing

In Chapter 3 issues were discussed regarding the possible switching between routing protocols as network conditions change. These issues, in light of the observed simulation results, shall now be considered further. Simulation results on the performance of flooding and minimum hop algorithms in the given network model subject to increasing intensity of change, show how the minimum hop algorithm degrades with increasing rates of change. Flooding is clearly robust to these topology changes but its inefficiency is very significant. This is especially true if the network is reasonably

stable.

Previous discussion in Chapter 3 suggested conditions when switching between minimum hop and flooding protocols could allow greater network throughput to be achieved by reducing routing costs. If intensity of change can be determined, then in the case of very dynamic conditions nodes should switch to flooding. Similarly, in more static environments nodes should switch to minimum hop routing. Moderate rates of change in topology make these decisions and associated gains less certain. For example, if traffic loads are light a preferred approach could be to utilise spare network capacity for minimum hop routing table maintenance.

These arguments are based on the assumption that routing table update overheads increase with increasing intensity of change in a network, to an extent where overheads exceed the routing cost of flooding. Under such circumstances, utilising a flooding protocol for routing purposes should improve network throughput. It is expected that update message overheads should increase proportionally with increases in intensity of change. This is because the table update algorithm for the minimum hop routing procedure is deterministic and event driven, with update message generation totally dependent on link cost change events. This has been verified through simulation experiments by monitoring the number of update messages generated for given intensities of change.

However, flooding can consume large quantities of available network resources, as is shown in Figure 5.1 for constrained flooding. It is unlikely that in this particular model, routing table update overheads could exceed the duplicated traffic load created by constrained flooding, even for very high levels of link cost change intensity. So the possibility of circumstances existing where gains in performance could be achieved by switching protocols becomes questionable.

An outcome from the investigation of minimum hop routing performance under dynamic conditions indicated that update traffic overhead only contributes a proportion of the total overhead traffic load. The dominant cause in degrading network performance is an increasing rate of end user retransmissions as intensity of change increases. Figure 5.8 illustrates this where high levels of change intensity result in decreased throughput and higher average delays. Routing table uncertainty where frozen nodes set their next node entries to be empty ($n_d(v) = \emptyset$) are the prime cause

of these retransmissions. This is due to packet discarding when no successor node can be identified. These subsequent high levels of user traffic retransmissions reflect poor network availability and illustrate the algorithm's inability to track high rates of topology change.

This issue of excessive retransmissions suggests that high levels of routing table update traffic loads should not be considered a prime factor in adopting a protocol switching strategy, instead failure of the minimum hop routing procedure to maintain routing tables under more dynamic conditions becomes the dominant factor. Consider situations where the intensity of dynamic behaviour varies both in time and over different regions of a network. Under quasi-static network conditions where intensity of change levels are relatively low, minimum hop routing operates efficiently. If conditions become more dynamic, flooding based routing may prove the only alternative if routing table convergence becomes too slow. This was shown in Figure 5.8 for the simulated network when conditions were such that if $\mathcal{I} > 0.04$ minimum hop routing essentially failed. Constrained flooding would be required whenever network operational conditions existed where intensity of change exceeded this threshold, resulting in poor routing efficiency and low throughput.

Deciding when to switch protocols then requires an indicator from which the intensity of change can be derived or inferred while still being suited to a decentralised implementation. From an individual node's perspective, intensity of change could be inferred by monitoring the rate of change to routing table entries for individual destinations. Conceivably this information could be used to switch all network nodes to flooding for those destinations affected, or switch only those nodes located in an unstable region of the network.

An immediate implication arising from the above discussion is in determining the actual intensity of change where the network becomes too dynamic. An absolute threshold for intensity of change would be dependent on a network's topology (size and shape) and most likely the nature of its dynamic behaviour. Moreover, it is difficult to conceive how nodes that have already switched to a flooding based routing procedure can decide when network conditions have stabilised sufficiently so that they can switch back to minimum hop routing, and resume participation in the table update procedures. There are no longer available the same indicators of

intensity of change with flooding because no routing tables are maintained, although monitoring the levels of adjacent link cost change frequency is a feasible test of local stability.

It is possible for nodes that are routing by constrained flooding to switch back to minimum hop routing, if that region of the network can be assessed as becoming stable again. This can be achieved by triggering adjacent nodes into resuming participation in routing table update procedures in the same way the algorithm is initialised.

Observations of the network model under simulation clearly illustrated the difficulty in selecting suitable threshold levels for switching decisions. Even with a single topology under well controlled dynamic conditions, there was no strong correlation between routing table rates of change and the identities of nodes that should route using flooding.

In addition to the issue of determining actual thresholds for intensity of change where protocol switching should occur, is the detrimental effect of flooding nodes ceasing their participation in the minimum hop routing table update procedures. This causes degraded routing performance at neighbouring nodes. If many nodes in the network are flooding and not replying to update messages, the minimum hop algorithm can conceivably deadlock and not adapt to further topology changes. Routing tables will not reflect true shortest paths, or perhaps even valid paths, to destinations.

Multiple protocol routing attempts to reduce update traffic overheads by switching to a flooding scheme so that routing tables are no longer required. It appears though, that update traffic overhead is not the dominant problem, and even if protocol switching is attempted to address the true problem of routing table uncertainty, actually finding a suitable technique for a general distributed switching strategy would be quite complex and raise concerns regarding its robustness. In addition, complications in applying this routing strategy appear likely to further degrade network performance if many nodes are not participating in table update procedures. For these reasons the hybrid routing procedure was considered a more viable option by focussing on the inability of minimum hop routing to track high rates of topology change. A variant of constrained flooding is then applied under well defined and

highly restricted conditions, leaving routing table update procedures to continue amongst all nodes.

5.3 Performance of the Proposed Hybrid Routing Scheme

Enhancing the minimum hop routing procedure by incorporating a localised version of the constrained flooding algorithm has greatly improved the routing performance in very dynamic conditions. If a network node is required to make a routing decision in order to forward a user or acknowledgment packet, and no knowledge of a successor node is currently available, the node invokes a local broadcast by transmitting the packet on all outgoing links. If the node is a transit node, the packet is not transmitted out on the same link from which it was received. Copies of the original user or acknowledgment packets effectively traverse multiple paths through the network towards their destination. Subsequent down tree nodes simply forward packets using next node entries in their respective routing tables, that is providing $n_d(v) \neq \emptyset$. Otherwise, these subsequent nodes also use local broadcasts. This results in a very high probability that user and acknowledgment packets will be received by destination nodes thereby reducing the frequency of user retransmissions. Network availability is greatly enhanced enabling much higher user traffic throughput.

By utilising the same sequence numbering scheme adopted by the constrained flooding algorithm, any looping packets are automatically discarded. This does not mean the algorithm is loop free. However, it assures that if any loops do occur, the packets concerned are removed. Therefore, any packets reaching their final destination do so along a loop free path. Conceivably a packet subject to a local broadcast could have all duplicate packets discarded before they reach the destination, due to packet looping. While this scenario is unlikely, it is possible. This implies that there is no absolute guarantee that packets will reach their final destinations. Local broadcasting though does provide a high probability of success whereas the minimum hop routing procedure itself does not provide any guarantee of delivery.

Figure 5.10 demonstrates the performance of the hybrid routing procedure as intensity of change increases. Even with very high intensities of change, the hybrid

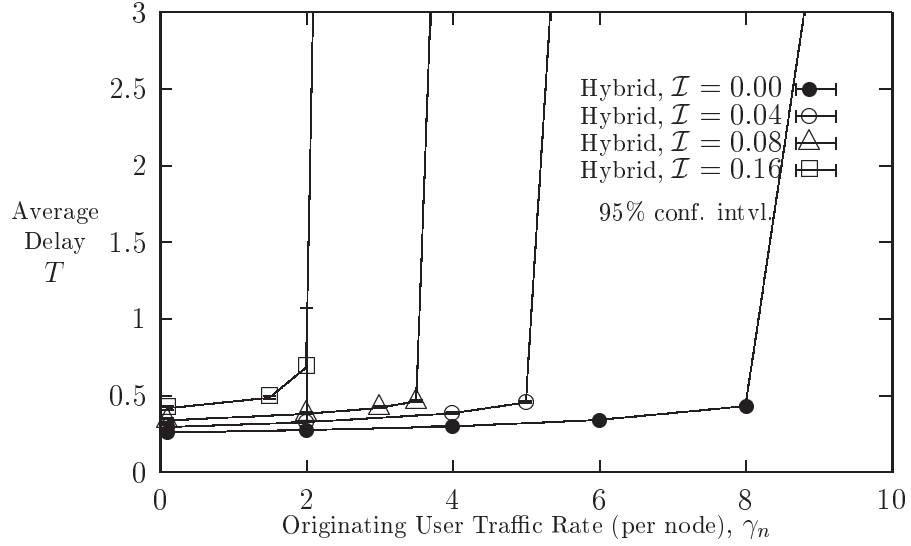


Figure 5.10: Delay using hybrid routing with increasing intensity of change

scheme exhibits the same characteristic threshold behaviour as previously shown for the M/M/1 queueing model. Throughput decreases as intensity of change increases due to the large amounts of routing table update traffic and flooding traffic with more instances of $n_d(v) = \emptyset$ occurring. Under very dynamic conditions throughput is tending towards that of pure flooding but the efficiency of this approach is significantly higher in terms of throughput when compared to constrained flooding alone. From previous simulation results, average throughput for constrained flooding and hybrid routing when intensity of change $\mathcal{I} = 0.16$ is 0.2 and 1.5 packets per second per node respectively. Overall the effect of this hybrid routing procedure is a smooth transition from a pure shortest path routing procedure in static and quasi-static conditions towards a flooding procedure as the network becomes more dynamic. By maintaining shortest path routing where possible, overall network performance is enhanced.

When operating hybrid routing, network capacity is consumed by routing table update traffic and duplicated user and acknowledgment packets caused by the local broadcasting strategy. These overheads reduce available throughput for user traffic and represent the routing cost for this particular routing procedure. An approximate comparison between hybrid and minimum hop overhead routing costs is illustrated in

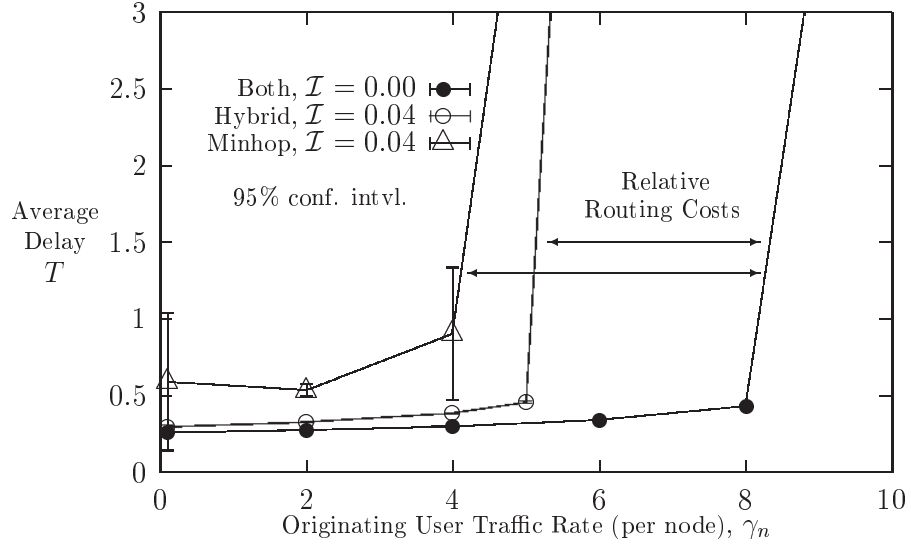


Figure 5.11: Relative routing transmission overheads

Figure 5.11 for $\mathcal{I} = 0.04$. For the same intensity of change the minimum hop routing procedure suffers higher overheads further reducing throughput when compared to hybrid routing. These costs are made relative to the case for a static network where maximum throughput is achieved.¹

In the case of minimum hop routing, overheads are dominated by routing table update traffic and user traffic retransmissions. Given that update traffic will be the same in both minimum hop and hybrid routing, due to identical network topology and dynamic conditions, it can be concluded that local broadcasting overhead is less than retransmission overheads when $\mathcal{I} = 0.04$. This is essentially why the hybrid routing procedure operates with lower routing costs. When $\mathcal{I} = 0.04$ the routing cost difference between minimum hop and hybrid routing are not substantial with minimum hop retransmission rates apparently only slightly exceeding local broadcasting overheads. As $\mathcal{I}$ increases though, hybrid routing demonstrates superior performance capabilities.

Figure 5.11 does not accurately reflect the true network capacity consumed by the routing procedures. In modelling dynamic behaviour, the degree of topology change is set to $\mathcal{D} = 0.05$. This implies that on average 5% of links are failed at

¹that is when using minimum hop fixed routing rules

any one time reducing total network capacity by approximately that same amount when compared to the static network with all links operational. Furthermore, the minimum hop algorithm does not adapt to traffic loads and the onset of congestion, so as topology changes traffic is diverted to alternative links with no regard to current link loading. It is very likely that saturation loads occur for lower rates of arriving user traffic as a result of this problem. Least cost routing would improve this situation by better managing link utilisation.

A significant factor affecting relative routing procedure efficiency is the ratio of user packet size to update packet size. This becomes an issue if the ratio decreases because update traffic loads will become much more significant and pure flooding relatively less inefficient. It may well prove that if user packet size is of the same order as update packet size, pure flooding becomes a more realistic option for routing in dynamic situations. In general though, user packet size will always be larger than routing table update message size even in cases where “user” packets represent call establishment packets in support of connection oriented services. This is due to the storage requirements of header fields in “user” packets compared to the efficiencies possible with update procedures by amalgamating multiple update messages into single packets.

To enable a more visible comparative assessment of the three routing procedures under varying dynamic conditions, routing power is determined for the maximum operating points of all three routing procedures when subjected to the given intensities of change. Throughput is calculated using Equ. (5.5) with values for probability of successful transmission derived from simulation results. The results are tabulated in Appendix A, Table A.1, and plotted in Figure 5.12. Hybrid routing clearly offers highest maximum operating throughput over the range of static through dynamic network environments. Operation with reasonable throughput under very dynamic conditions gives an indication of the algorithms robust nature. Flooding demonstrates poor efficiency but is also robust in dynamic topologies. The relative inefficiency of flooding can be attributed to the ratio of user packet to control packet size being set at 16:1. Note that the minimum hop algorithm degrades until it essentially fails as indicated previously in Figure 5.8.

Desirable properties of the Jaffe-Moss shortest path algorithm and constrained

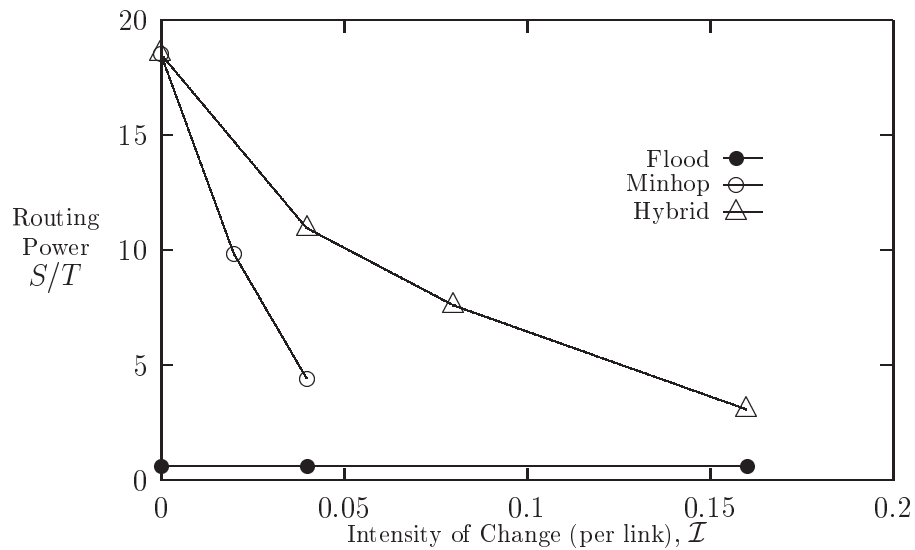


Figure 5.12: Comparative performance over varying intensity of change

flooding are preserved in this hybrid routing procedure. Rapid convergence, minimal update traffic load and loop free routes to the destination are guaranteed through the use of sequence numbering, while local broadcasting also provides a high assurance of successful transmission under very dynamic conditions. Disadvantages include the extra complexity and storage requirements within network nodes to manage the sequence numbering of user packets and the increased packet overhead required to contain these sequence numbers. These disadvantages are however outweighed by the high throughput/delay performance and self adaptive properties exhibited by the algorithm.

The combined hybrid scheme offers very significant improvements in performance when compared to pure flooding. In addition, by exploiting flooding characteristics using the local broadcasting strategy to directly address the routing table uncertainty problem, minimum hop routing performance is enhanced as the frequency of link cost changes increase. This is achieved by successfully reaching the destination nodes with high probability on first attempts. The focus of this strategy is to incorporate complementary advantageous features of generalised least cost and flooding type algorithms into a more flexible routing strategy capable of adapting to changing operating conditions. The resulting hybrid scheme is relatively simple,

robust and very efficient in dynamic networking environments.

5.4 Effect of Network Size and Connectivity

It is expected that the performance of flooding, minimum hop and hybrid routing will be dependent on specific network topologies. The number of nodes and links together with the degree of interconnection dictate a network's total capacity, its number of source-destination pairs and the choice of available paths. These parameters directly affect the operation of routing procedures, the amount of control and duplicated traffic that is generated and able to be sustained within the network and are reflected in the procedures' resultant performance. Of particular interest is whether hybrid routing performance remains superior to the performance of other schemes when operated in different network configurations.

A network topology can be described by its size (the number of nodes), its average degree of connectivity (the number of links to adjacent nodes) and its shape. Obviously these parameters give rise to an infinite array of possible topologies so a restricted subset is again required for the purpose of performing a limited number of simulation experiments. Given the even distribution of traffic loading admitted to the networks under investigation, network shape tends to be related to the degree of connectivity. For this reason the network shape shall be constrained to that of a Manhattan grid network. To explore variations in size and connectivity while still limiting the number of simulations, two additional topologies shall be studied. One topology has the same size (64 nodes) as that of Figure 4.2 but twice the connectivity (i.e. average degree 8 containing 256 bidirectional links) and the second topology has the same connectivity (degree 4) but twice the network size (128 nodes). Note that the 64 node degree 8 topology is not strictly a Manhattan network, however, the additional links interconnect adjacent nodes diagonally maintaining the toroid shape.

Both these new topologies have in common the same total number of links. This number of links is exactly double the number of links in the original 64 node degree 4 network investigated previously in this chapter. As a result, for any given intensity of change $\mathcal{I}$, the 64 node degree 8 and 128 node degree 4 networks are subjected to

twice the number of average link cost changes per second occurring in the network. Additionally, in the 64 node degree 4 network the degree of topology change was set at $\mathcal{D} = 0.05$. This same proportion of average number of links down at any given time is maintained for the two new network topologies introduced.

Simulation results, for the 64 node degree 8 network, of delay throughput performance at the maximum operating points for various intensities of change are tabulated in Appendix A Table A.2. The subsequent derived routing power versus intensity of change is plotted in Figure 5.13. Note that a similar trend has resulted in the curves representing all three routing procedures as was shown in Figure 5.12 for the 64 node degree 4 network. Flooding demonstrates similar delay throughput performance (refer tables A.2 and A.1) due to its broadcasting mechanism. Even though the degree 8 network has more links, flooding simply duplicates additional packets to be transmitted on these extra links.

Minimum hop routing shows signs of degrading performance with increased intensity of change as expected. However, not only did retransmissions dominate overall performance for $\mathcal{I} = 0.06$, as was the case shown in Figure 5.8, but the delay throughput curve for $\mathcal{I} = 0.04$ also resulted in wide variances in output statistics. For this reason no maximum operating point was determined for $\mathcal{I} = 0.04$. Perhaps this further reduction in performance with the degree 8 network can be explained by having twice the number of links, resulting in twice the average number of link cost changes occurring per second in the network for the same intensity of change. The minimum hop routing procedure is effectively working twice as hard to maintain routing tables within network nodes for the 64 node degree 8 network given the same intensity of change.

Hybrid routing also degrades in performance as intensity of change increases but remains operational even under very dynamic conditions (e.g. $\mathcal{I} = 0.16$). Figure 5.14 compares hybrid routing performance for the 64 node degree 4 network with the 64 node degree 8 network. Since the degree 8 network has twice the connectivity, total network capacity is doubled with greater probability of alternative paths being available improving throughput performance by reducing the average number of hops. This is reflected in Tables A.1 and A.2 showing increased throughput and reduced average delays for the degree 8 network. It is interesting that the 64

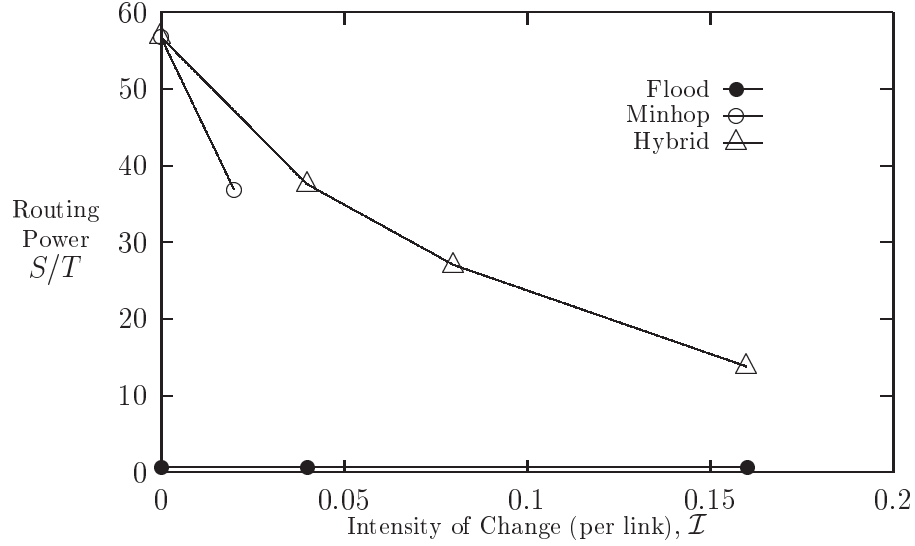


Figure 5.13: Comparative performance - 64 nodes with connectivity of degree 8

node degree 8 network offers significant improvement in performance for intensity of change $\mathcal{I} = 0.16$ over the degree 4 network, even though the average rate of link cost change events occurring in the network is doubled. This is most likely attributed to the extra connectivity reducing the impact of link cost change events.

Whereas increasing network connectivity improves overall performance, increasing network size appears to reduce network performance as perceived by individual users. Performance of the larger network, 128 node degree 4, is shown in Figure 5.15 derived from simulation results tabulated in Appendix A Table A.3. Throughput, average delay and subsequent routing power metrics are all scaled on a per node basis to reflect quality of service performance experienced by the individual nodes (users).

Flooding continues to demonstrate its independence from changes in topology, but absolute performance is approximately 1/3 that of the 64 node degree 4 and degree 8 topologies. Reduced throughput and increased delays are clearly the cause of this as indicated in Tables A.1, A.2 and A.3. With double the number of nodes flooding the network, it is not surprising that each individual node's throughput has effectively been reduced proportionally.

Minimum hop routing again degrades with increased intensity of change and fails

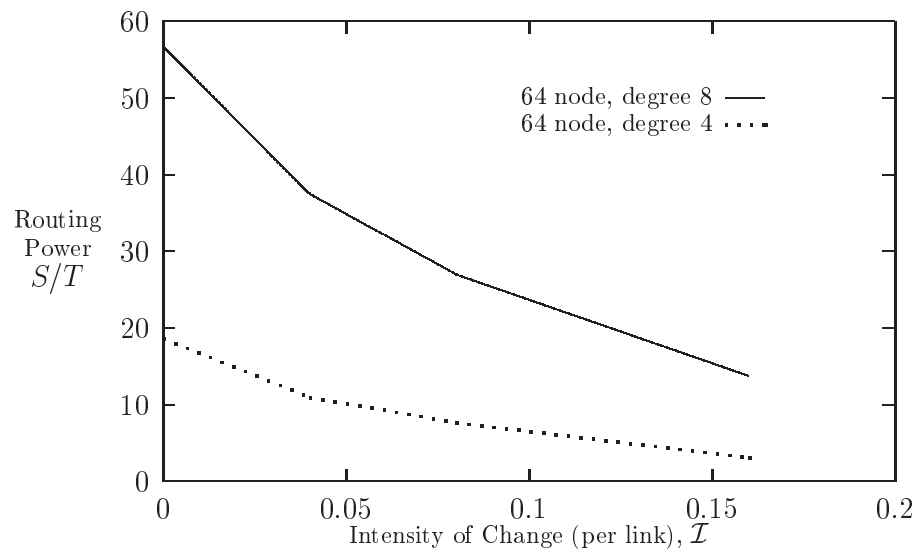


Figure 5.14: Hybrid routing performance with increased connectivity

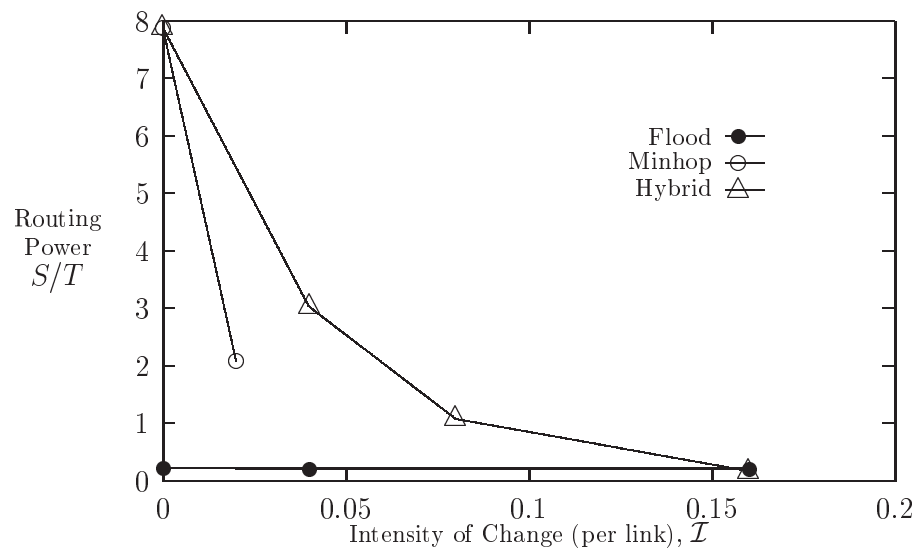


Figure 5.15: Comparative performance - 128 nodes with connectivity of degree 4

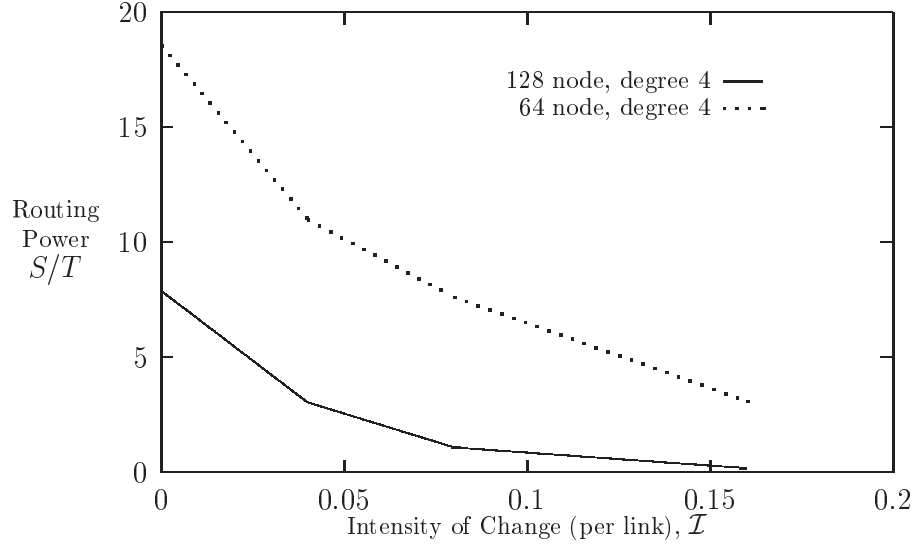


Figure 5.16: Hybrid routing performance with increased network size

to operate effectively if $\mathcal{I} > 0.02$. As with flooding, absolute routing power per node is reduced in the 128 node network with the increased number of nodes resulting in lower per node throughput (γ'_n) at the maximum operating points, refer to the tables in Appendix A. Furthermore, with the 128 node network the average number of hops has increased, causing higher average delay that subsequently results in decreased routing power.

The performance curve of hybrid routing illustrated in Figure 5.15 is consistent with previous simulations demonstrating gradual degradation as intensity of change within the network increases. Again, decreased routing power is attributed to the overall increase in network size. Figure 5.16 clearly illustrates the reduced performance as perceived by individual users when operating in the 128 node degree 4 network as compared to the original network configuration.

Of particular interest in Figure 5.15 is that when operating hybrid routing with an intensity of change $\mathcal{I} = 0.16$ in the 128 node degree 4 network, routing power is less than that of flooding. This results primarily from high average delay with performance being significantly worse than both the 64 node topologies, see Table A.3 and compare flooding and hybrid routing when $\mathcal{I} = 0.16$. Simulation results tend to indicate that in this large 128 node network subjected to very dynamic

conditions,¹ the hybrid routing procedure is contributing similar routing costs to that of constrained flooding. This identifies a limit in network configuration and corresponding operational conditions for the hybrid routing procedure. However, in reality this represents very dynamic conditions with 0.16 average link cost changes per second per link occurring in a network with 256 bidirectional links equating to an average of 82 link cost change events occurring per second within the entire network.

¹Recall that twice the average number of link cost change events per second are occurring in the network compared to the 64 node degree 4 configuration

Chapter 6

Modifications to Hybrid Routing

Chapter 5 demonstrated performance gains from operating hybrid routing in a network model subjected to varying dynamic conditions. While the hybrid scheme offers significant improvements over flooding, with high intensity of change (e.g. $\mathcal{I} = 0.16$), high routing costs are incurred by the broadcasting process limiting available throughput for user traffic. As discussed, total routing costs can be attributed to minimum hop routing table update traffic, duplicated user traffic flows resulting from local broadcasts, and perhaps retransmissions in the event of some lost user traffic. Recall that user traffic can be discarded if broadcast packets are directed along paths containing loops, or discarded by a failed link flushing an output queue. A question arises as to whether it is possible to enhance network performance using alternative strategies to further reduce routing costs.

This chapter explores two techniques targeting the further reduction of routing costs associated with hybrid routing. The first approach seeks to reduce the amount of duplicated traffic resulting from the local broadcasting strategy. Instead of flooding a user packet on all outgoing links, if no valid next node entry exists in the routing table, utilise existing knowledge contained in a node's own distance table and forward the packet along a next best choice by accepting the uncertainty of outdated information. A second strategy is aimed at speeding up convergence time performance of routing tables by transporting routing table update traffic at a higher priority than user traffic throughout the network. Improving routing table recovery time should reduce the number of initiated local broadcasts and hence the routing cost. Added complexity is of course introduced in the network to manage

this prioritised traffic.

6.1 Next Best Path Strategy

In the event that no valid next node exists in a node's routing table (i.e. $n_d(v) = \emptyset$), hybrid routing broadcasts copies of the user packet onto every outgoing link. Conceptually this is in response to having no knowledge of an appropriate successor node so all neighbouring nodes are attempted. Generally $n_d(v) = \emptyset$ because the node v is currently in a freeze state for the destination d due to an adjacent link failure. However, even if a node is in a freeze state, its distance table is continually updated by setting $C_d(v, w) = c + l(v, w)$ every time an update message, $\text{MSG}(d, c, 0)$ or $\text{MSG}(d, c, 1)$, is received. It is therefore reasonable to conclude that even though no minimum next node has been calculated for the routing table, a reasonable near minimum cost next node entry may exist in the distance table.

A next best choice routing procedure is proposed that calculates a next node offering the least cost path to a required destination from the distance table. This least cost next node is then used as the successor node for forwarding a user packet if no valid next node entry exists in the routing table. An advantage of this scheme is that only one outgoing link (the most likely) is chosen for forwarding user traffic, instead of broadcasting on all outgoing links as is done with hybrid routing. Routing overheads are reduced by not broadcasting, hopefully enabling higher network throughput. Also, the chosen link has a higher probability of success than an approach based on random link selection.

This approach can be considered similar to using a basic decentralised Ford-Fulkerson algorithm with Jaffe-Moss coordination to speed convergence time and reduce update message traffic loads. Refer to Table 2.1 for performance comparisons of these algorithms. Of course, like Ford-Fulkerson's algorithm, the next best choice routing procedure is no longer guaranteed loop free. Loop free routing is a key feature of the Jaffe-Moss algorithm and the original motivation for development of the Coordinated Update Procedure.

Since the next best choice scheme no longer guarantees loop freedom, a sequence numbering procedure (derived from constrained flooding) is required to discard any

packets that traverse a looped path. This ensures that any packets successfully received do so along a loop free path. As was the case for hybrid routing, discarded packets will lead to retransmissions.

Hybrid routing can offer multiple chances that an originated user packet will reach its destination due to the local broadcasting strategy. This reduces the likelihood of retransmissions occurring. However, the next best choice scheme forwards traffic along a single path only and one packet being discarded will definitely lead to a retransmission. For this reason, the next best choice scheme must be considered less robust than the hybrid scheme. Although robust performance is deemed to be a prime requirement of routing procedures used in dynamic networks [95], [79, p. 342], the next best choice routing strategy will be examined because its true capability is uncertain.

6.1.1 Delay throughput performance

Delay performance for the next best choice routing procedure, operating in a network subject to increasing intensity of change, is illustrated in Figure 6.1. The network model that was simulated consisted of 64 nodes with 4 degrees of connectivity and an average of 5% links failed at any given time ($\mathcal{D} = 0.05$). This remains consistent with the basic experimental conditions used in Chapter 5. Figure 6.1 shows a set of characteristic performance curves consistent with expected threshold type behaviour, and clearly illustrates the onset of congestion occurring with lower originating user traffic loads as intensity of change increases. Routing costs are contributed by routing table update message traffic, retransmissions and routing via suboptimal paths. With a very dynamic network topology ($\mathcal{I} = 0.16$), the maximum operating point suggests that relatively high throughput can be achieved under these demanding conditions.

Probability of successful transmission using the next best choice routing procedure with selective repeat ARQ is plotted in Figure 6.2. These curves correlate well with those of Figure 6.1 and demonstrate that the probability of successful transmission is close to 1 at the maximum operating points. Throughput is therefore reasonably well represented by the originating user traffic load, γ_n up to γ'_n .

Results presented in Figure 6.1 for a static network ($\mathcal{I} = 0$) show the maximum

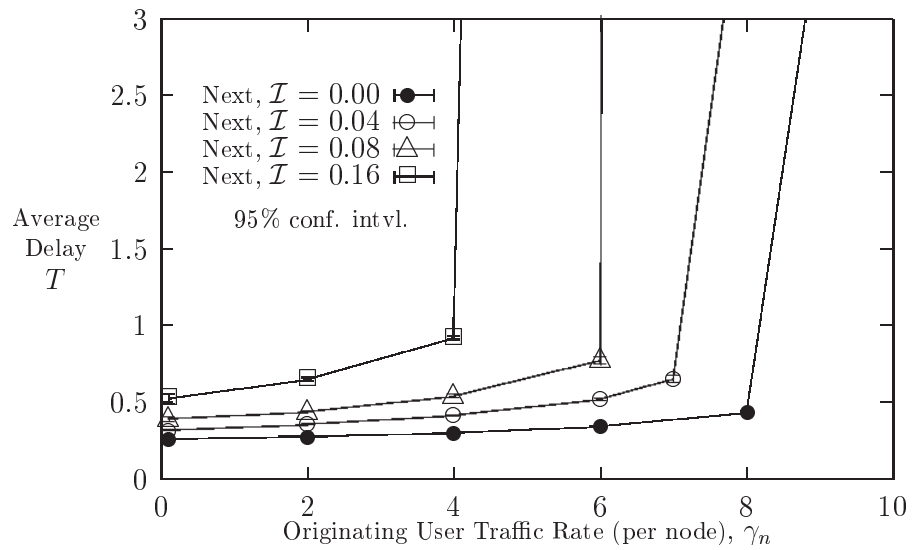


Figure 6.1: Delay using next choice with increasing intensity of change

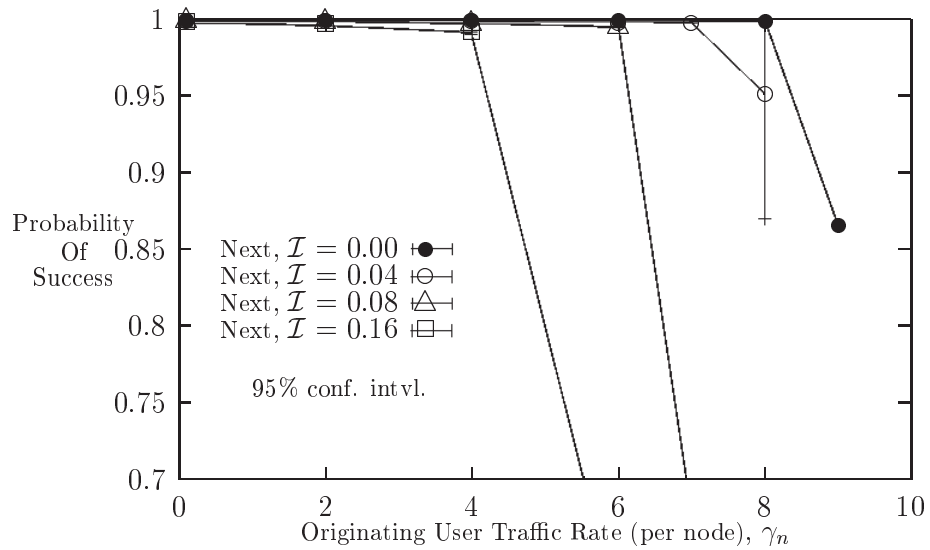


Figure 6.2: Probability of success using next choice with increasing intensity of change

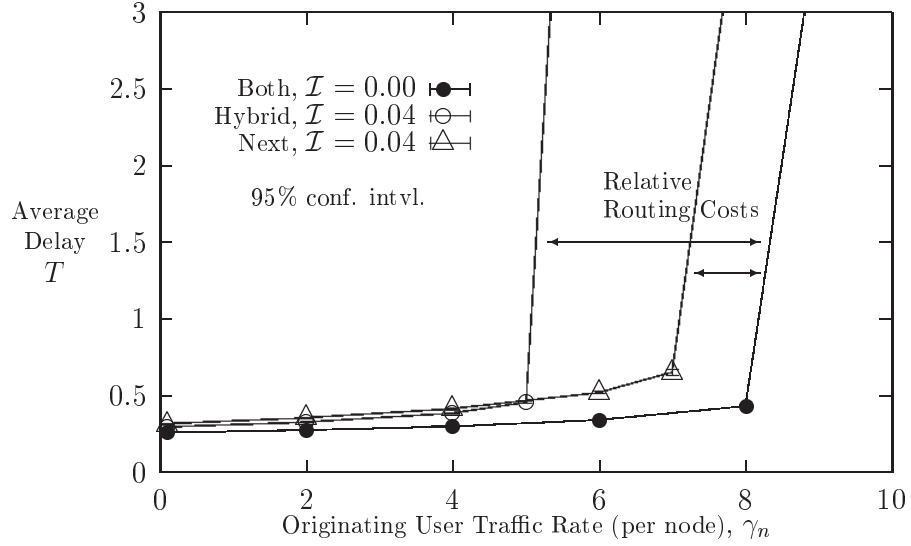


Figure 6.3: Relative routing transmission overheads

operating point to be $\gamma'_n = 8$. When $\mathcal{I} = 0.16$, γ'_n is reduced to 4. Under these dynamic conditions, almost 50% of network capacity is consumed by routing overhead costs. At the maximum operating point when $\mathcal{I} = 0.16$ and $\gamma'_n = 4$, there were on average 153419 user packets sent, 151875 packets received and 14361 lost packets retransmitted. This illustrates that on average a 10% retransmission rate is required to maintain the probability of successful transmission at 0.99. This high retransmission rate is reflected in the relatively high average delay at the maximum operating point for this curve.

6.1.2 Comparison with hybrid routing

Figures 6.1 and 5.10 show the performance of next best choice and hybrid routing respectively, under the same dynamic conditions. On examination of these curves, a first impression is that next best choice routing significantly out-performs the hybrid routing procedure. Figure 6.3 highlights the approximate relative routing cost overheads associated with the two routing procedures when intensity of change $\mathcal{I} = 0.04$.

However, the maximum operating point is a function of delay as well as throughput. Table 6.1 has been derived from the simulation results presented in Table A.1,

Table 6.1: Throughput and Delay Comparison with Hybrid Routing

$\mathcal{I}$	Throughput, S		Delay, T	
	Hybrid	Next	Hybrid	Next
0.00	7.99	7.99	0.43	0.43
0.04	5.00	6.98	0.46	0.65
0.08	3.50	5.97	0.46	0.77
0.16	1.50	3.96	0.49	0.92

Appendix A, and compares throughput and delay for both routing procedures subject to increasing intensity of change. From the results contained in Table 6.1, it becomes clear that throughput is indeed higher when using the next best choice scheme, although average user delay for this routing procedure is also much higher. Higher retransmission rates and longer transit paths are the most likely causes of this increase in delay. As a result, delay performance of the next best choice routing procedure tends to degrade overall network performance when performance is assessed as routing power.

A comparison of flooding, minimum hop, hybrid and next best choice routing procedures are shown in Figure 6.4. Simulation results used to derive this figure are tabulated in Table A.1, Appendix A. Under these simulation conditions, the next best choice routing procedure does not offer any significant gains over hybrid routing. As intensity of change increases, the next best choice scheme sustains higher throughput than hybrid routing but traffic delays increase more rapidly. The probability of successfully reaching a destination in one attempt, with next best choice routing, is greatly reduced when compared with hybrid routing. As a result, average users experience higher “no load” delays due to increased retransmissions. The reduction in overhead by not flooding with next best choice routing does not significantly compensate the reduced probability of first attempt success. In addition, hybrid routing is more likely to select the least cost path by broadcasting, if routing tables are uncertain, while next best choice routing is totally reliant on the current state of distance tables if no valid next node entry exists in the routing table.

Overall performance of the next best choice algorithm, expressed in routing

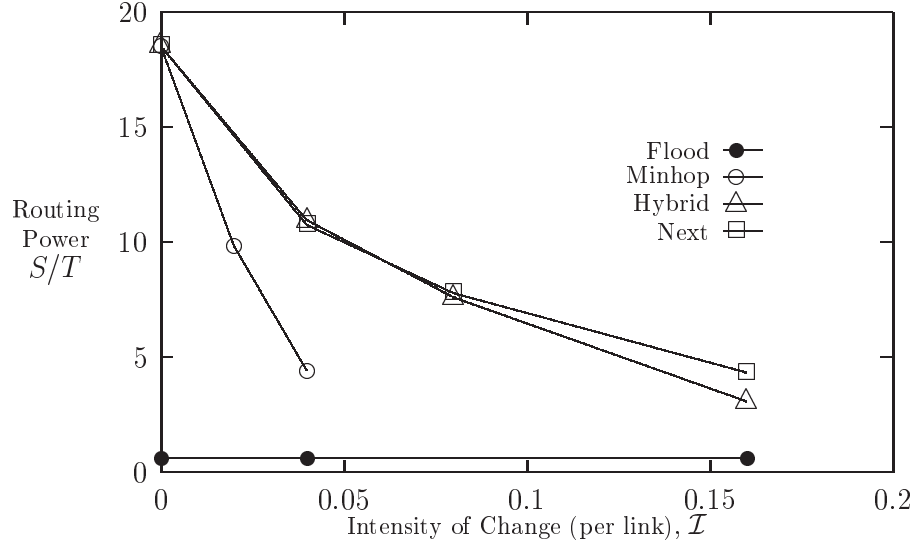


Figure 6.4: Comparative performance over varying intensity of change

power, is comparable to the performance of hybrid routing. For $\mathcal{I} = 0.16$, the next best choice scheme actually outperforms hybrid routing slightly (see Figure 6.4). However, the next best choice scheme relies on distance table information to indicate a possible route during periods when there is no successor node entry contained within the routing table. This can lead to high retransmission rates, due to lost traffic (refer to page 112), because the next best choice routes are subjected to the same routing uncertainties as the minimum hop route. So, given that this simulated network model is reasonably well connected and the average proportion of failed links is small,¹ it is reasonable to expect that the next best choice routing procedure should perform relatively well.

Hybrid routing should in general be more robust than the next best choice scheme due to its local broadcast protocol under conditions of routing table uncertainty. However, it is not clear from the preceding results whether the next best choice scheme could achieve significantly superior results under different circumstances. Even though the next best choice strategy could be explored further, to ascertain its relative performance characteristics more closely, a more detailed study of hybrid routing has been adopted instead. This is due to the lack of significant

¹Recall $\mathcal{D} = 0.05$, i.e. on average 5% of links are failed at any time.

performance improvements evident in next best choice routing when compared with hybrid routing, under the conditions simulated, and the consideration that hybrid routing should in general be more robust.

6.2 Prioritised Routing Table Updates

Results presented in Chapter 5 indicate that routing table uncertainty is a dominant factor in causing user retransmissions that reduce shortest path routing procedure performance under dynamic conditions. Intuition suggests that performance could be improved through increased speed of routing table convergence by transmitting routing table update messages at a higher priority than user traffic. The faster convergence of routing table information resulting from minimisation of control traffic delays should consequently reduce instances of invalid next node entries, i.e. $n_d(v) = \emptyset$. For minimum hop routing this could reduce the amount of lost traffic and subsequent retransmissions. In the case of hybrid routing, the extent of local broadcasting may be reduced providing more efficient routing.

If high priority control traffic receives preferential treatment, it is likely that low priority user traffic may experience long delays if there are excessive quantities of control traffic generated within the network. This situation could be expected under dynamic conditions where high routing table update traffic loads are encountered. From previous results it appears that update traffic loads in the range of dynamic conditions investigated are substantial, making intuitive assessments of priority performance difficult. In the following section a study is conducted using the simulation model to accurately reflect update traffic loads.

6.2.1 The priority mechanism

A nonpreemptive priority queueing discipline is adopted where packets are queued according to their respective priority classes. These classes are strictly separated based on the priority level that packets belong to. There are two priority classes used in this model, a low priority class for user packets and a high priority class including routing table update messages and user acknowledgment packets. The high priority control packets are always queued ahead of low priority user packets

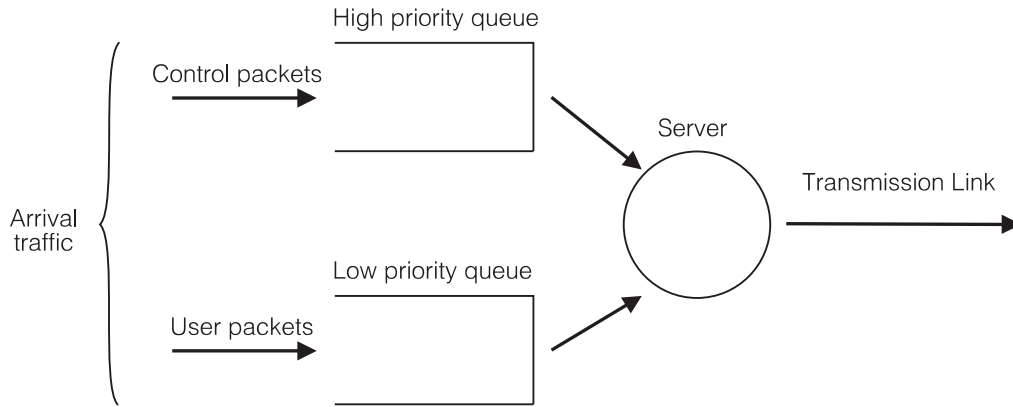


Figure 6.5: Priority queueing model for packets transmitted on link l

but do not preempt any low priority user packet already in service [82, p. 119], [4, p. 61].

Figure 6.5 depicts the nonpreemptive priority queueing system for an outgoing transmission link l in the network model. Arriving traffic comprises user packets (placed in the low priority queue) and control packets including routing table update and acknowledgment packets, and user acknowledgment packets (all placed in the high priority queue). High priority packets are immediately serviced with low priority packets being transmitted onto the outgoing link only when the high priority queue is empty.

Priority is implemented in the simulation model by assigning a priority attribute to every packet before forwarding it to an outgoing queueing system. In the simulation model, each link is represented as a single queue with a given packet being inserted in the queue ahead of all packets of lower priority and behind all packets of equal or higher priority. Packets are then simply serviced by always transmitting onto a link the packet at the head of a queue.

For a particular priority class, the expected waiting time in a queue is only dependent on the number of packets of equal or higher priority present in the queueing system. Waiting time is unrelated to the number of lower priority packets queued. Therefore, if the proportion of high priority control traffic is light compared to the

low priority user traffic load, this priority mechanism can be expected to significantly reduce control packet waiting times while only increasing user delays by a small amount. This result can be demonstrated analytically for an M/G/1 nonpreemptive priority queue [4, p. 62]. However, it can also be shown that lower priority traffic can experience excessive, unstable queueing delays under increased loads of high priority traffic [82, p. 121].

6.2.2 Performance comparison using priority

Nonpreemptive priority queueing is simulated using the 64 node network with 4 degrees of connectivity and compared with network performance achieved by the same model without using the priority mechanism. Results for the no priority case are illustrated in Figure 5.12. Delay and throughput measurements used to derive routing power are extracted from user packet statistics only, this is to represent users' perceived quality of service. Delays experienced by control traffic have not been recorded but are of course indirectly reflected in the user statistics. The performance of constrained flooding is not being discussed here because no control traffic is associated with this protocol.

Minimum hop routing procedure performance is plotted with and without nonpreemptive priority in Figure 6.6. Simulation results used for this figure are tabled in Appendix A Table A.1 for no priority and Table A.4 with high priority assigned to control traffic. In both cases when $\mathcal{I} = 0.06$ output statistics exhibited wide variances due to retransmission delays as was illustrated in Figure 5.8 for the case of no priority. The application of priority appears to offer no improvement in minimum hop routing performance and actually slightly degrades routing power for the intensities of change simulated.

Hybrid routing using nonpreemptive priority produces a result similar to that of minimum hop routing. That is, no substantial gains are evident in using priority and this is illustrated in Figure 6.7. This common lack of any significant performance gains in both routing procedures must be attributed to the large quantities of routing table update traffic, generated under the high intensities of change used in the simulation experiments, causing excessive waiting times in the low priority queues. Consider that even "low" intensities of change, such as $\mathcal{I} = 0.02$, represents an

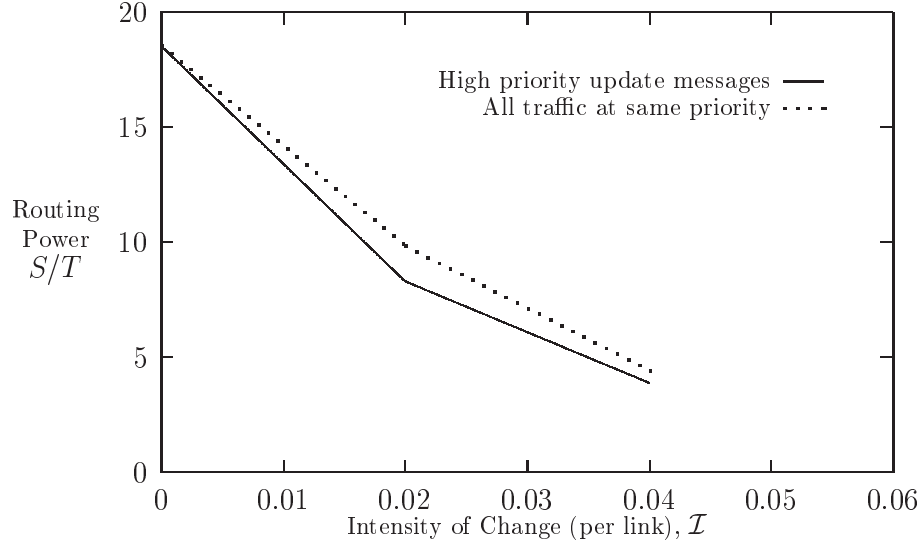


Figure 6.6: Minimum hop routing with prioritised table updates

average of 5 link cost changes per second occurring in this network.

The effect of excessive traffic load in high priority queues is shown in Tables 6.2 and 6.3 derived from the simulation results recorded in Appendix A. Both average delay and throughput decrease between the experiments assigning high priority to control traffic and equal priority for all packets, in both hybrid and minimum hop routing. As a result, the routing power ratio does not change significantly and this is reflected in the network performance curves, Figures 6.6 and 6.7. This implies that in the case of using priority, user packets experience substantial reduction in available network capacity due to large quantities of control traffic ahead of user traffic in queues. User packets that are able to reach their destinations experience improved delay performance due to more efficient routing with improved routing table convergence times.

From the above results it appears that nonpreemptive priority improves average delay performance at the expense of throughput performance, refer Tables 6.2 and 6.3. Therefore the net result, as reflected in routing power, over all intensities of change raises doubts on the usefulness of such an approach under varying dynamic conditions, see Figures 6.6 and 6.7.

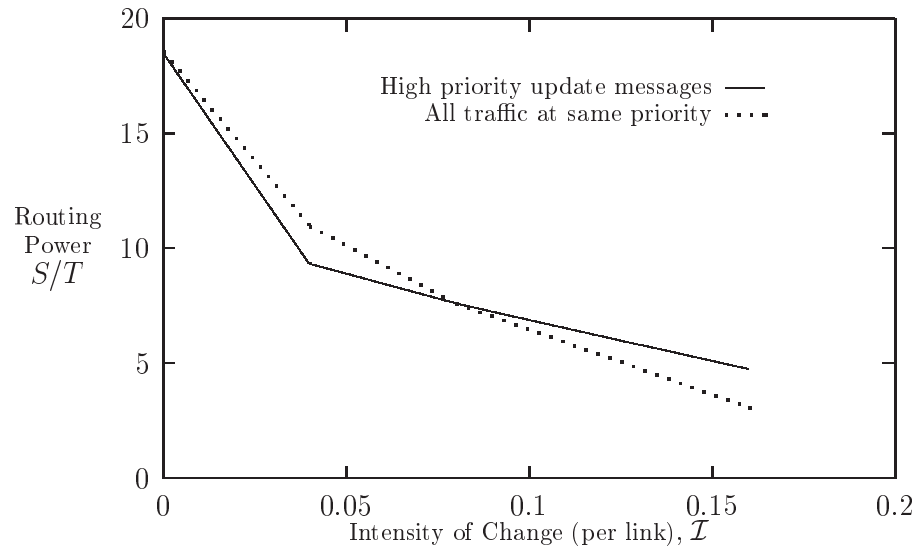


Figure 6.7: Hybrid routing with prioritised table updates

Table 6.2: Comparison of Minimum Hop Routing with and without Priority

$\mathcal{I}$	Throughput, S		Delay, T	
	No Priority	Priority	No Priority	Priority
0.02	5.47	4.48	0.56	0.54
0.04	3.97	1.99	0.90	0.51

Table 6.3: Comparison of Hybrid Routing with and without Priority

$\mathcal{I}$	Throughput, S		Delay, T	
	No Priority	Priority	No Priority	Priority
0.04	5.00	2.98	0.46	0.32
0.08	3.50	2.50	0.46	0.33
0.16	1.50	1.50	0.49	0.32

Chapter 7

Verifying Performance Robustness of Hybrid Routing

Results obtained from simulation experiments described in Chapters 5 and 6 have demonstrated significant performance gains from using hybrid routing over minimum hop routing and constrained flooding in a network subjected to varying degrees of topology change intensity. The network model was exercised with evenly distributed random link changes in order to gain greater insight into averaged simulation output statistics. While this was achieved satisfactorily, further consideration must now be given to the performance of these same routing strategies under more realistic, coordinated patterns of topology change. This will provide an indication of how robust hybrid routing is when used under general conditions.

7.1 Expected Routing Algorithm Performance

By its very nature, constrained flooding is always able to reach a destination node within a network under any form of dynamic network behaviour, provided that there exists a path through the network to that destination. Flood search routing can only fail to reach a destination if the destination node is physically isolated from other parts of the network (i.e. the network is partitioned). Of course the price for this robust performance is constrained flooding's inefficiency, limiting available network throughput.

As shown on page 50, the Jaffe-Moss minimum hop algorithm is triggered solely

by adjacent link cost change events or through receiving routing table update messages from neighbouring nodes. All possible events are catered for through the various steps the algorithm performs. These include node failure, node isolation, nodes coming up on the network and network partitioning. The clear deterministic behaviour of this algorithm dictates exactly how it will behave when subjected to any series of events. Although the algorithm requires quasi-static network conditions to maintain reasonable throughput performance, the algorithm can continue execution in a stable deterministic fashion even under very dynamic conditions.

It is expected that the Jaffe-Moss minimum hop routing procedure will continue reacting to link cost change events as defined by its algorithm under any form of dynamic topology change behaviour. However, as found in Chapter 5, high rates of change within the network will inevitably lead to high occurrences of unknown successor nodes in routing tables causing user traffic to be discarded. The reduction in throughput performance resulting from this effect is not caused by the algorithm failing to operate in a deterministic fashion, but rather the inability of the algorithm to keep up with rapid changes occurring in the network.

Hybrid routing is constructed by combining the minimum hop and constrained flooding algorithms together. The minimum hop routing procedure is allowed to continue execution unhindered while the local broadcasting strategy supplements routing decisions when no successor node is identified within routing tables. It is expected that this technique will cope with any topology change scenario due to the deterministic behaviour previously described for minimum hop routing and the robust simplicity of constrained flooding. Although hybrid routing will not provide optimum routing decisions, it should operate correctly under any dynamic conditions and can be anticipated to offer superior throughput delay performance over the use of minimum hop routing or constrained flooding alone.

7.2 Verification Through Simulation Modelling

While intuitive arguments for the robust performance of hybrid routing can be stated, some form of verification is required to gain confidence in these arguments. As discussed previously in this thesis, simulation modelling will be required to cap-

ture the complexity of routing table update procedures operating under dynamic network conditions. To avoid performing vast numbers of simulation experiments, some specific scenarios have been chosen to investigate the robust performance capabilities of hybrid routing. These scenarios include large and rapid changes to network topology, regional and time varying dynamic behaviour, and correlated patterns of topology change.

7.3 Large Changes to Topology

Degree of topology change ($\mathcal{D}$) has been maintained relatively small in previous simulation network models in order to reduce the complex effects caused by network partitioning.¹ This approach enabled clearer observations and subsequent understanding of derived simulation results. By using the same network models and further increasing the degree of topology change parameter, $\mathcal{D}$, it is possible to create large changes in topology shape throughout the duration of the simulation experiment.

For this simulation experiment, the network model is a 64 node degree 4 Manhattan network as described in Chapter 4 and illustrated in Figure 4.2. No priority queueing mechanisms have been applied to any of the network models investigated in this chapter of the thesis. Degree of topology change is set at 0.25 representing an average 25% of links that can be in the failed state at any particular time throughout the simulation, i.e. $\mathcal{D} = 0.25$. To reflect static through dynamic operational conditions, the intensity of change $\mathcal{I}$ is varied accordingly.

Under dynamic conditions causing frequent link failure and link recovery events, the shape of the network topology will clearly alter drastically over time. Additionally with such a high proportion of links down, it is highly probable that the network may be partitioned with nodes possibly becoming isolated. These severe network conditions operating under high intensities of change will effectively test the robust characteristics of hybrid routing.

¹Degree of topology change $\mathcal{D}$ has previously been set to 0.05, representing an average maximum of 5% links failed at any given time.

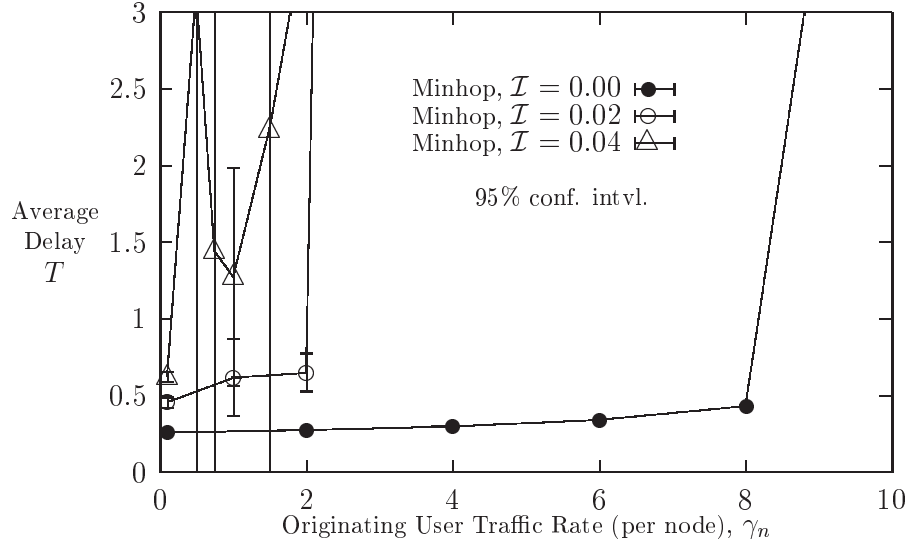


Figure 7.1: Minimum hop routing with increased degree of topology change

7.3.1 Results of network simulation

Delay versus throughput performance for minimum hop and hybrid routing are illustrated in Figures 7.1 and 7.2 respectively. When operating minimum hop routing with $\mathcal{I} = 0.02$, threshold behaviour is exhibited in the delay throughput curve. However, beyond this intensity of change, probability of first transmission attempt success decreases with greater dominance of retransmissions affecting end-to-end delay performance. This is shown in Figure 7.1 for $\mathcal{I} = 0.04$.

Hybrid routing demonstrates clear threshold type behaviour even for much higher intensities of change (refer Figure 7.2). Although when $\mathcal{I} = 0.16$ the no load delay is significantly higher representing high routing costs due to routing table update procedures and duplicated user traffic created by the local broadcasting strategy.

Both minimum hop and hybrid routing appear to only support low user throughput's when operating under nonstatic conditions (i.e. $\mathcal{I} > 0$) relative to the static case. This is caused directly by the nature of the dynamic behaviour modelled in the simulation. Consider that when $\mathcal{D} = 0.25$, 25% of links are down on average when the simulation is operating in statistical equilibrium. This becomes a significant proportion of overall network capacity that is removed whenever $\mathcal{I} > 0$, even if intensity of change is very small. Moreover with significantly fewer links carry-

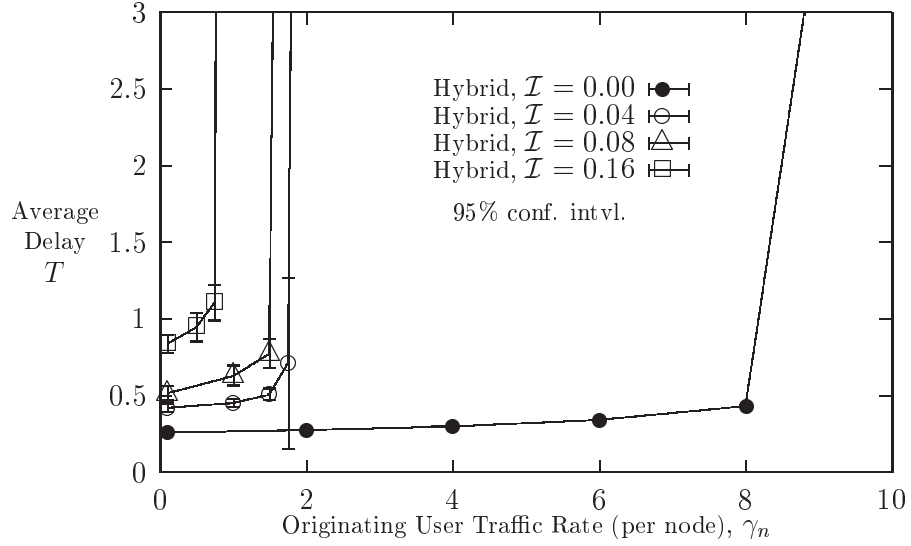


Figure 7.2: Hybrid routing with increased degree of topology change

ing the offered traffic load, it is reasonable to expect a much higher probability of localised congestion occurring because no traffic overload control is undertaken by the shortest path routing algorithm. That is in its current implementation as a minimum hop routing procedure.

In addition, with so many links failed in the network, it is highly likely that the network becomes partitioned with nodes being isolated throughout the simulation. Since nodes will attempt transmissions to all other nodes in the network uniformly, and some destination nodes may become isolated, then discarded user packets will invoke retransmissions further loading the network while reductions in the probability of successful transmission will occur. The combined impact of all these effects greatly reduces useful user throughput performance between static and dynamic conditions, even when intensity of change is very small.

A comparative assessment of constrained flooding, minimum hop routing and hybrid routing subjected to this increase in degree of topology change ($\mathcal{D} = 0.25$) is illustrated in Figure 7.3. Results obtained from the simulation model and used to derive this figure are tabulated in Appendix A Table A.5. Generally the results are as expected. Flooding remains robust irrespective of topology change patterns while minimum hop routing degrades in performance until it essentially fails as conditions

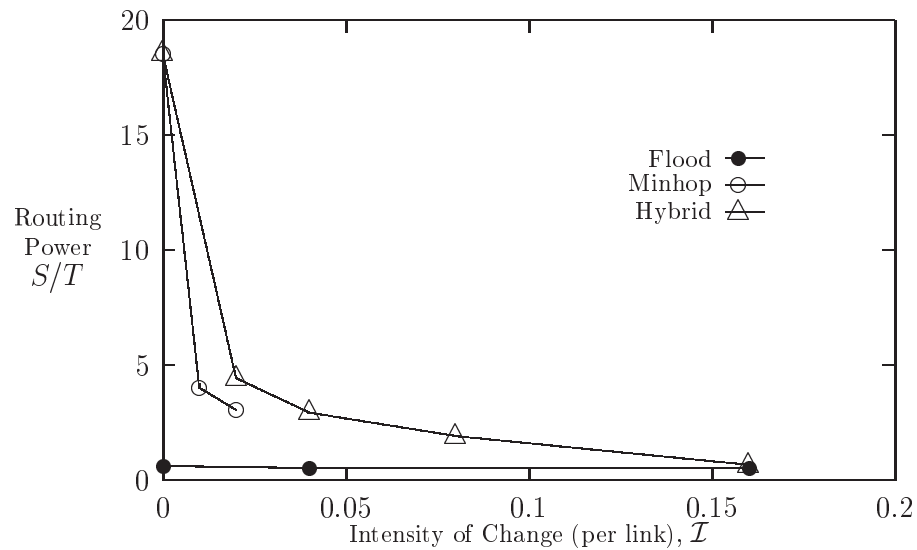


Figure 7.3: Comparative performance with increased degree of topology change

exceed those of a quasi-static nature. Hybrid routing offers improved performance over the entire range of dynamic conditions when compared with flooding and minimum hop routing. However, overall performance of hybrid routing is quite poor but as explained previously this is due to the severe nature of the link cost change behaviour assumed in the model.

7.4 Regional and Time Varying Link Cost Change Intensity

All dynamic network behaviour examined so far has occurred uniformly across the entire network. Of particular interest is whether hybrid routing operates consistently under conditions of regional or localised change, and when levels of dynamic network behaviour vary over time. Such a capability can readily be demonstrated using the existing network model by restricting the number of links within the network whose state can be altered by link cost change events to a small region of the topology. If this region of dynamic links is further allowed to move across the network topology during execution of a simulation, effects of time varying dynamic behaviour can also be modelled.

The same 64 node, 4 degree connectivity, Manhattan network model as described in Chapter 4 is again used for this simulation experiment. Recall from Chapter 4 that for a given intensity of change $\mathcal{I}$, links are scheduled for failure and later restoration by deriving simulation event times from two probability distributions. When simulation time arrives to fail a link, the link is randomly selected from all possible links within the network G . This process is repeated until the chosen link is one that is not currently in a failed state. By simply placing an additional constraint, namely the link must also reside within a specified region of the topology, a region of changing link costs can be simulated.

For the purpose of this simulation experiment, the region is defined as an area encompassing 20 nodes with a total of 31 links being susceptible to link cost change events. Degree of topology change is set at $\mathcal{D} = 0.05$ corresponding to an average of 5% links failed at any particular time within the network. So within the region of 31 links, this represents an average of 6 links failed at any given time. Intensity of change $\mathcal{I}$ is varied over a set of simulation experiments to assess effects caused by increasing dynamic network conditions.

In order to incorporate elements of time varying behaviour, the region described above moves across the network topology during simulation run time. This is shown in Figure 7.4 with the region being highlighted by the shaded area. Movement of the region across the network topology is in the direction of the arrow (as shown in the figure) and occurs in discrete steps during execution of the simulation. In addition to regional and time varying dynamic conditions, this pattern of topology change introduces some correlated aspects to the dynamic behaviour moving away from the pure stochastic conditions modelled so far.

7.4.1 Simulation results

Simulation experiments modelling a region of dynamic link costs moving across the network yielded the following results. Minimum hop routing behaved as expected with delay throughput curves deviating from a classic threshold type of response to wide variances in output statistics as intensity of change increases. This is due to excessive quantities of lost traffic and can be seen in Figure 7.5.

Figure 7.6 illustrates the delay throughput performance of hybrid routing sub-

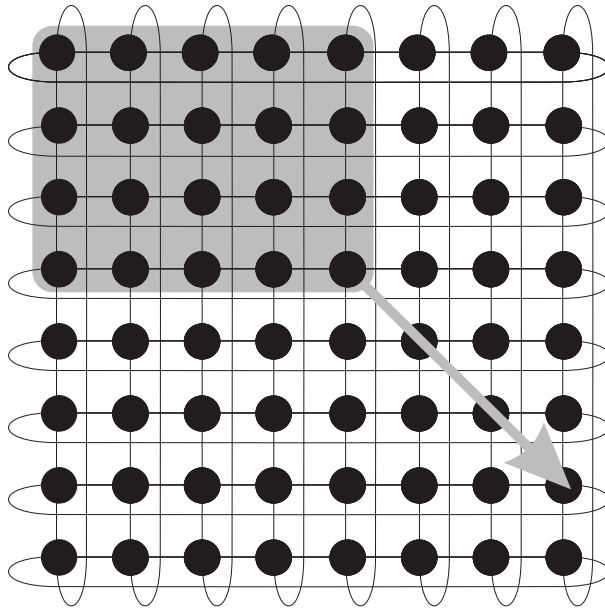


Figure 7.4: Region of dynamic change (shaded area) moving across the network

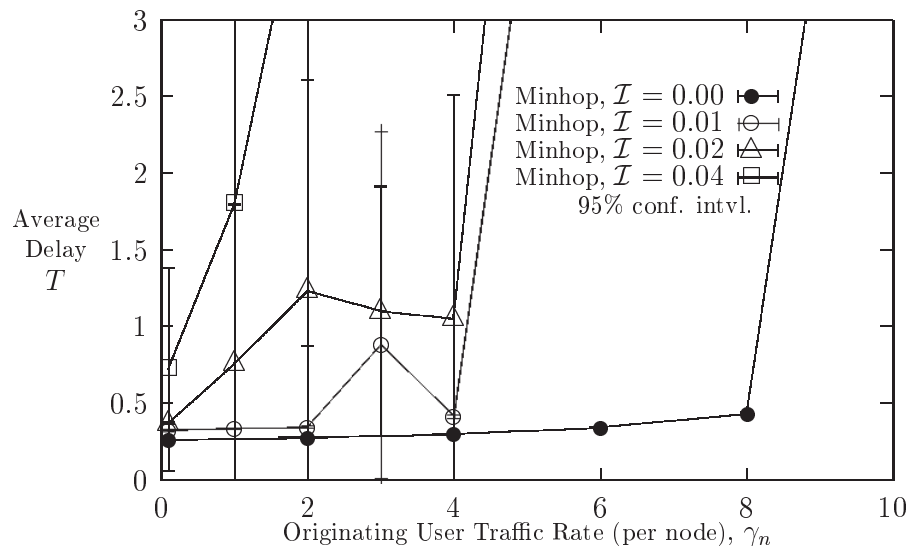


Figure 7.5: Minimum hop routing with regional topology change

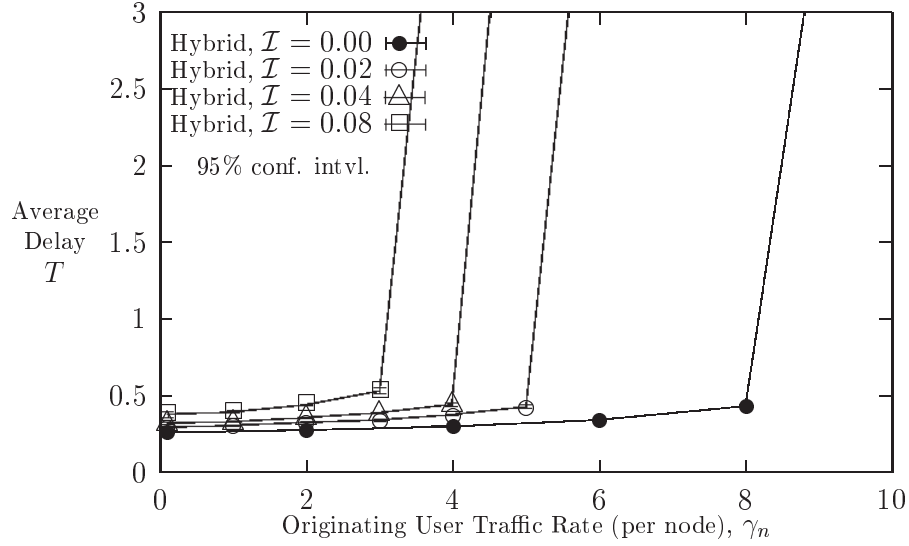


Figure 7.6: Hybrid routing with regional topology change

jected to regional topology change. Note that consistent threshold type characteristics are evident even for higher intensities of change ($\mathcal{I} = 0.08$). Similar to the previous model of increased degree of topology change, Figure 7.6 shows a nonlinear relationship between maximum operating point and intensity of change. Although routing table update traffic increases proportionally with the rate of link cost change events occurring (resulting in a proportional reduction of available capacity), other factors such as the number of retransmissions, user traffic load distributions and effects caused by the flushing of output queues after link failure, further complicate total routing overhead characteristics.

In this model of regional change when the degree of topology change is $\mathcal{D} = 0.05$, small values for intensity of change again result in the same average number of links failed at any time, but under such circumstances links remain down for longer periods of time. This represents a much smaller proportion of lost network capacity when compared to the previous dynamic model. However, traffic congestion caused by network bottlenecks is certainly a significant issue for this model. This is because while there are only relatively few links failed at any particular time, all failed links are concentrated within a small region of the entire network topology. This results in the shortest path algorithm determining minimum hop paths around a “hole” in

the network with no regard to traffic loading distributions. Recall from Equ. (2.5) that severe congestion in any link can dominate overall delay performance.

Another factor constraining throughput performance also relates to the concentrated nature of this regional change model. Parameters such as intensity of change $\mathcal{I}$ and degree of topology change $\mathcal{D}$ are averaged over the entire network. However in the case of this dynamic model with regional change, $\mathcal{I}$ and $\mathcal{D}$ are constrained to a small region effectively greatly increasing the local $\mathcal{I}$ and $\mathcal{D}$ values as perceived by nodes near this region of change. This results in a much higher probability of unknown successor nodes in routing tables for nodes executing the shortest path routing procedure near this region of dynamic link cost change behaviour.

Comparative performance of the three routing procedures are shown in Figure 7.7. Simulation results and subsequent calculations of routing power used to derive this figure are tabled in Appendix A Table A.6. Note that while minimum hop routing fails to track high intensities of change, even when the change is localised, hybrid routing offers significantly superior performance over flooding under very high intensities of change. This efficient robust result is achieved with hybrid routing because local broadcasting is only initiated around the region of change, and more efficient shortest path routing is used throughout the more stable regions of the network. The ability of hybrid routing to utilise shortest path routing tables in stable regions of the network is the key to its efficient operation, and this dynamic model illustrates its inherent capability of distributed adaptation to changing conditions.

7.5 Topology Change in a Mobile Radio Network

In Chapter 4 the simulation model was developed with reference to a mobile packet radio network. However, much of the complexity associated with such a network was abstracted out of the model to enable a greater understanding of the operation of the minimum hop and hybrid routing procedures. Given that behaviour of these routing procedures under dynamic conditions is now well understood, it is appropriate to return to a more realistic model of a mobile packet radio network.

By specifying the movement of mobile nodes over an arbitrary terrain, it is

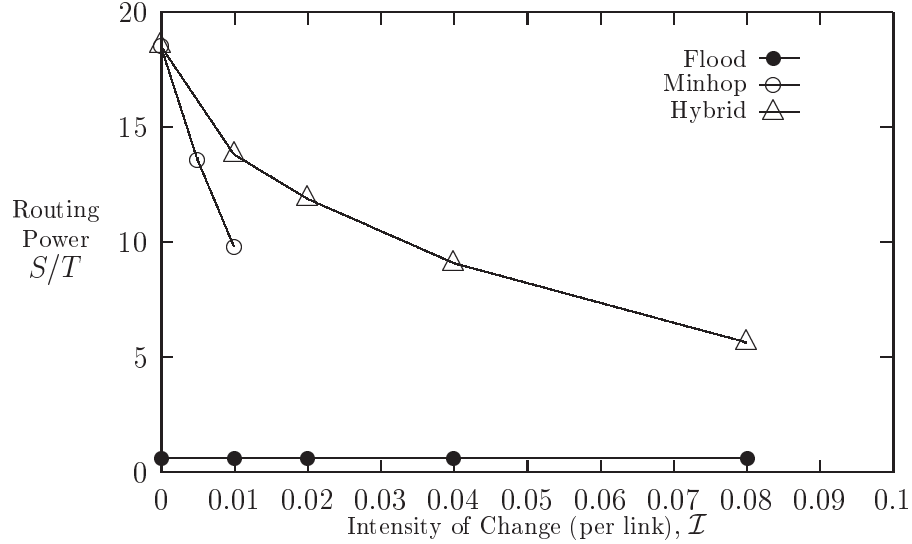


Figure 7.7: Comparative performance with regional topology change

possible to introduce correlated and intelligently coordinated dynamic behaviour into the simulation model. In addition, by utilising a completely different simulation model in a manner such as this, improved confidence in the robust performance of hybrid routing can be achieved further verifying the operation of this adaptive routing strategy.

In the model, nodes move across the surface of a smooth Earth in both latitude and longitude at a fixed altitude. While in line-of-sight range, nodes communicate with each other using simulated radio transmission equipment. A centralised topology control entity tracks the motion of all nodes maintaining a global topology database and regularly informs individual nodes of the identity of nearest neighbours. Distance and routing tables within these nodes can then be updated with the establishment of new communication links. The distributed shortest path algorithm is executed as before in a completely decentralised manner using the interconnecting transmission links for all internode communication.

7.5.1 Mobile packet radio node model

Functions performed by the mobile nodes and modelled here include sourcing and sinking of user packets, switching of transit user traffic and executing the rout-

ing procedures as described previously in Chapter 4. Although, instead of being interconnected to other nodes via fixed communications links as done in previous simulation models, in this network model radio transmission links are used.

Internode links are modelled as logical multichannel radio links supporting a maximum of six simultaneous full duplex channels. This allows each node to interconnect with six neighbours forming an overall network topology of maximum degree six. Each channel is capable of sustaining a bidirectional data rate of 16kbps remaining consistent with link models used previously. Transmission and reception of radio signals is assumed through an isotropic radiating antenna of 0dB gain in all directions. Receivers are modelled as ideally matching transmission bandwidth requirements and possessing a noise figure of 1, that is no noise introduced to the receive path by the receiver itself. While nodes are within line of sight range, it is assumed that modulation, line coding and error control coding schemes sufficiently compensate for channel noise and distortion such that all transmissions are successfully received without error. Interference between network nodes is avoided by the assumption of an existing underlying multiple access scheme.

Antenna position for the isotropic radiator is set at an altitude of 3m above the latitude, longitude coordinate of the node it is servicing. With the antennas of two nodes at this altitude, line-of-sight operational radio range is approximately 13km [108, p. 26-13]. This is over a smooth Earth surface of effective radius $4/3$ the Earth's actual radius. The effective Earth radius allows for propagation of the radio signal beyond the Earth's visible horizon, caused by variations in the atmosphere's refractive index with altitude. To ensure that only the topology control entity dictates when nodes come in and out of range, an artificial maximum range is set at 10km. In this way the 3km range buffer ensures that two nodes slightly less than 10km apart and moving directly away from each other at high speed, will not lose radio contact between topology control update cycles.

7.5.2 Topology control

A topology control function is required to specify the neighbouring nodes for each individual node and form an overall network topology at any instance in time. Topology control must be performed before any shortest path routing algorithm can be

executed. Nodes can then be made aware of their respective closest neighbours so that communication links can be configured and new columns inserted into their distance tables for use by routing procedures. As nodes are free to move, the topology will change requiring updates to transmission link and distance table configurations within network nodes. So to maintain effective topology control, periodic updates of topology status are invoked. Within this model the topology control function is executed centrally to simplify implementation.

If node coordinates at any particular time are considered a set of points in two dimensions, then topology control as described here is attempting to solve the nearest neighbour problem. Problems of this nature can typically be solved using techniques of triangulation where all points are interconnected by nonintersecting straight line segments so that every internal region forms a triangle. Applications for triangulation methods include finite element analysis [109, p. 189] and terrain surface mapping [110].

One criteria that constitutes a “good” triangulation is that of maximising the smallest angles within the triangles [109, p. 190]. In other words construct all the triangles as equilateral in shape as possible by ensuring the minimum measure of angles within all triangles in a triangulation is maximised (the maxmin angle criterion) [111, p. 202]. The resulting triangulation is referred to as a Delaunay triangulation and defined “as the unique triangulation such that the circumcircle of each triangle does not contain any other point in its interior” [109, p. 223]. This property is illustrated in Figure 7.8. Triangulation computation for a set of N points in the plane can be constructed in $O(N \log N)$ time [111, p. 201].

A Delaunay triangulation represents a very desirable topology for a multihop packet radio network because triangulation edges tend to diverge from each node in widely separated directions. That is there are few “long and skinny” triangles present in the topology avoiding multiple radio links established in approximately the same direction [112]. Another general property of triangulations is that the average degree of nodes is less than six [112], this is the reason a maximum degree of six has been adopted for the simulation model.

As mentioned previously, topology control is performed centrally within the sim-

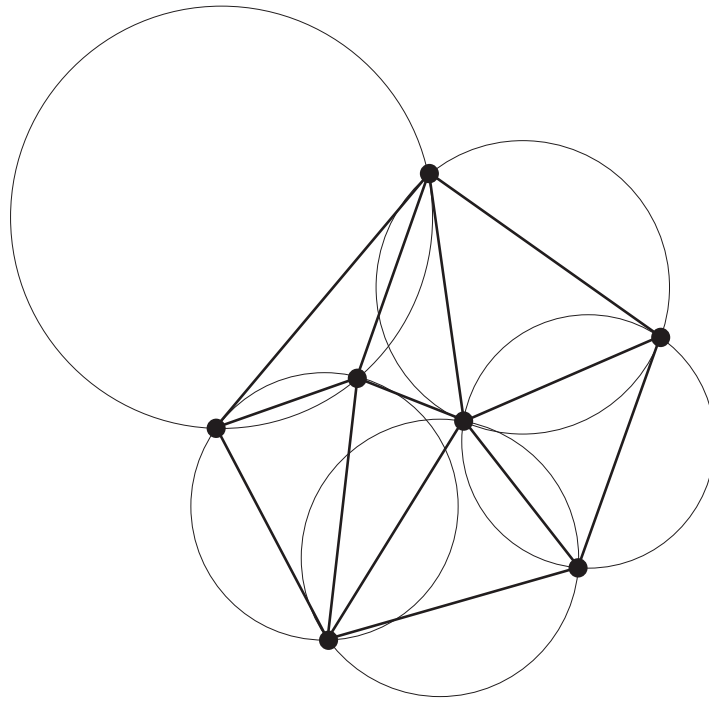


Figure 7.8: Delaunay triangles and some of their circumscribing circles

ulation model. At the end of each update interval,¹ topology control measures the current location of every node and constructs a Delaunay triangulation from the resultant set of node coordinates. Each edge length within the triangulation is measured, any exceeding 10km in length are assumed out of radio range and removed from the topology. In addition, any nodes with more than six edges linking neighbouring nodes also have their edges removed to return the nodes to a nodal degree of six. This results in a stable network topology, for the duration of the update interval, enabling the establishment of communication links between adjacent nodes as indicated by the remaining edges in the triangulation. Nodes are individually notified if their respective local topology has altered since the previous update interval.

One method for determining a Delaunay triangulation is to use the divide-and-conquer algorithm [110], [113, p. 402]. The divide-and-conquer algorithm recursively divides the data points into approximately equal subsets until each data set consists of a minimum of four points. Triangles are formed at this lowest level and these are then progressively merged together until the whole area is completed re-

¹A periodic update interval of 5 seconds is used within the simulation model as this is adequate to track maximum separation of nodes before they move out of respective range.

sulting in a Delaunay triangular mesh. Software developed by Jonathan Richard Shewchuk from the School of Computer Science at Carnegie Mellon University calculates Delaunay triangulations for a set of data points using the divide-and-conquer method. This software has been used with permission in the calculation of Delaunay triangulations for the purpose of topology control used within the simulation model investigated in this thesis. The actual software is available through the Internet from <http://www.netlib.org/voronoi/triangle.shar>. Decentralised algorithms for determining Delaunay triangulations for use in multihop packet radio network topology control have been proposed and demonstrated to be an efficient technique for general network topology control [112, 114].

7.5.3 Modelling a multiple access scheme

Central topology control periodically reallocates transmission links to nearest neighbours as required for every node within the network. Distance and routing tables are updated using this information together with the subsequent execution of the minimum hop routing procedure. Resultant routing tables can then be utilised to forward user packets to their appropriate successor nodes. Similarly for constrained flooding, nodes must be aware of neighbouring topology so packets can be broadcast to adjacent nodes via the nominated transmission links. An issue arises as to how individual nodes schedule the transmission of packets to these adjacent nodes. This occurs due to multiple nodes communicating with each other using omnidirectional antennas requiring a strategy to prevent nodes interfering with each other's radio broadcasts. In practical radio networks, multiple access schemes are generally employed to regulate the use of this shared medium. Before describing how this issue is resolved within the simulation model investigated here, general functionality of some multiple access schemes shall be examined to better appreciate the problem and understand the reasoning for the approach taken in this network model.

Frequency Division Multiple Access (FDMA) is a technique where individual channels are each assigned a separate frequency within a broader transmission spectrum. As communication channels are separated by frequency, interference is avoided. FDMA is a simple resource allocation mechanism typically employed in satellite communications to separate individual user channels where user terminals

are collocated within a single antenna footprint. However, complexity arises in implementing FDMA due to intermodulation interference caused by modulating two or more signals and transmitting these through a final nonlinear power amplifier[115, p. 582]. Furthermore, inefficient use is made of frequency spectrum if FDMA is used in radio broadcasting applications where independent frequencies are required to interconnect many nodes. Although this latter problem can be reduced by employing frequency reuse in different areas of the network that are beyond an interfering range.

An alternative to FDMA is Time Division Multiple Access (TDMA) where, as the name suggests, individual channel access is separated by time rather than frequency. This results in more efficient use of the frequency spectrum but requires synchronisation between all nodes so that individual nodes are aware of exactly when to transmit a packet without interfering with other broadcasts in the network [116, p.398].

A technique based on direct sequence spread spectrum called Code Division Multiple Access (CDMA) avoids these problems by directly modulating a data modulated carrier with a wideband spreading code. A receiving node demodulates this received signal using the same code as that used for the transmission. With unique codes used for different transmissions, multiple transceivers can communicate within the same frequency spectrum and at the same time. CDMA transmissions continuously interfere with each other, but over such wide bandwidths that signal to noise ratios at the receivers can be quite acceptable if well chosen spreading codes are used. However, if a third node operating a different spreading code is physically close to one of a pair of communicating nodes, this third node can act as a wideband jammer significantly interfering with the node that is further away. As a result, very good power control techniques must be employed in CDMA systems to overcome this problem [117, p. 532].

A multiple access scheme called Carrier Sense Multiple Access (CSMA) has been employed in packet radio networks [82, p. 395]. In its simplest form this technique listens to a broadcast channel to sense whether the channel is idle. If the channel is idle a packet can immediately be transmitted. If the channel is busy, the node must wait (persist) until the channel is free by continually monitoring the broadcast

channel. If more than one node is waiting for the channel to become idle a scheme such as this can result in collisions. An alternative scheme called Nonpersistent CSMA does not persistently wait until the channel is free, instead the node schedules a reattempt at some later randomly chosen time [3, p. 128]. At this new point in time the channel is again sensed and if free the packet can be transmitted. This reduces the probability of collisions occurring between multiple nodes detecting a busy channel. Throughput and delay performance degrades though as traffic load increases with these approaches.

Describing the above multiple access schemes highlights some of the issues to be considered when selecting access strategies for practical networks and the possible impact on overall network performance. The underlying objective of multiple access methods is to provide a scheduling mechanism for transmissions between multiple nodes by effectively establishing logical links that interconnect these nodes without mutual interference.

To avoid the complications of modelling a multiple access scheme and interpreting its subsequent distortion of network performance results, an idealised strategy has been modelled instead. Whenever a node desires to transmit a user packet or control packet to a neighbouring node, the node is provided resources such that communication occurs immediately without interference from other nodes. This avoids any delay or packet corruption being caused by a multiple access model. As described previously, the existence of all logical links to neighbouring nodes is defined by the topology control function.

7.5.4 Routing performance

A simulation program to investigate performance of the routing procedures, operating within the mobile packet radio network model, has been created using MIL 3's OPNET Radio Modeller simulation tool. Model libraries supplied with the tool were utilised to model radio channel characteristics and receiver performance. Individual routing procedures and user traffic models remain identical to those developed previously for network models described in Chapter 4.

Functionality of the new node model has been described in the preceding subsections while the simulated network model developed consists of 64 nodes positioned

within an area of approximately 3000 square kilometres. Nodes are each assigned a trajectory that defines their position at any time during the simulation. Nodes move along these trajectories linearly with time until they reach their respective trajectory end points at the end of simulation execution. Some nodes remain in a fixed position throughout the simulation. Many of the trajectories are identical in magnitude and direction to those of several other nodes within the network. This represents clusters of nodes moving together as a group during the simulation. Actual initial node locations, together with their predetermined motion trajectories, are illustrated in Appendix B. Also in Appendix B is a diagram showing an example Delaunay triangulation generated from initial node coordinates at the beginning of the simulation.

Only a single dynamic scenario has been simulated here because this is sufficient to achieve the overall objective of measuring relative routing procedure performance with a network subjected to general, very dynamic, intelligently coordinated and correlated patterns of change. Simulation duration is fixed at 600 seconds with the simulation model finite state machine executing approximately 17 million state transition events during this time for a single set of input parameters (e.g. γ_n). Even this relatively short duration results in very long computation times,¹ requiring the simulated time to be minimised and only sufficiently long enough such that output statistics are insensitive to any increase in simulation duration.

Simulation results depicting delay performance with increasing user traffic load for minimum hop routing are shown in Figure 7.9. From these results, it is clear that minimum hop routing is unable to track the dynamic changes occurring to the network topology. Note that the ordinate axis for this graph has an increased upper bound compared with other plots of delay response. Large variances in averaged output delay statistics for the various simulation experiments performed can be seen, these indicate effects caused by significant retransmissions resulting in average delays well beyond the expected “no load” delay. The “no load” delay equals the transmission delay² multiplied by the average number of hops. Retransmission

¹Execution speed is approximately 300 events/second using a Sun Microsystems Ultra Sparc workstation.

²Transmission delay is 0.064 seconds for a 1024 bit user packet transmitted over a 16kbps link.

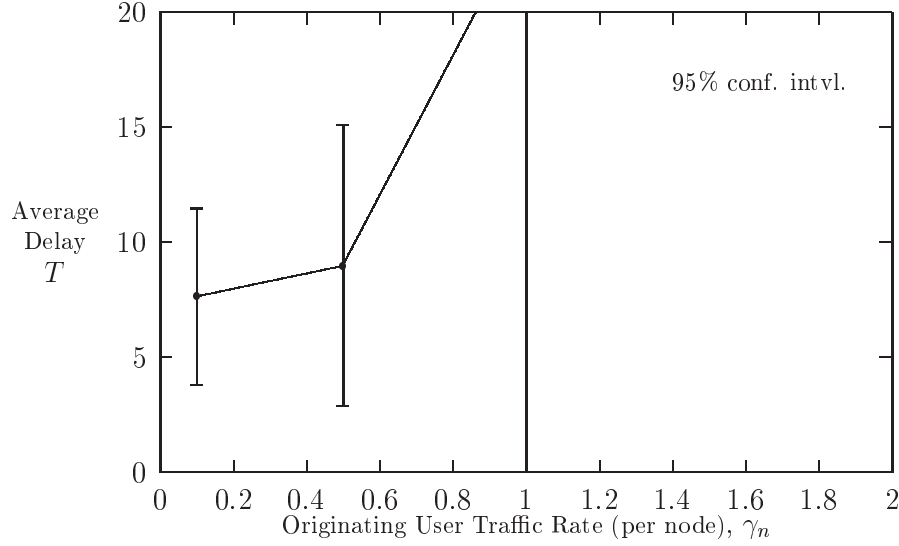


Figure 7.9: Minimum hop routing performance in the mobile radio network model

counts were monitored throughout these experiments and in the case of minimum hop routing, retransmission rates were found to be of the same order as original user packet transmission attempts. Furthermore, simulation results for flooding and hybrid routing, shown later, reveal that the actual “no load” delay corresponds to several hops through the network on average. This is reasonable when considering that an average hop count of 4 was previously determined for the Manhattan network of similar size.

Delay performance for constrained flooding is illustrated in Figure 7.10. The delay throughput curve is similar to previous results presented, again demonstrating the robust performance of this routing procedure. An interesting point to observe is that the maximum operating point, $\gamma'_n = 0.2$, is identical to previously presented models of the Manhattan network, e.g. the 64 node network with connectivity of degree 4. The dynamic mobile radio network model consists of 64 nodes and in the initial topology 146 interconnecting links. This is somewhat similar in topology size to the 128 links found in the Manhattan network and accounts for the similarity in results.

Hybrid routing performance is plotted in Figure 7.11 showing a clear “no load” delay threshold at the maximum operating point. Not only does this routing proce-

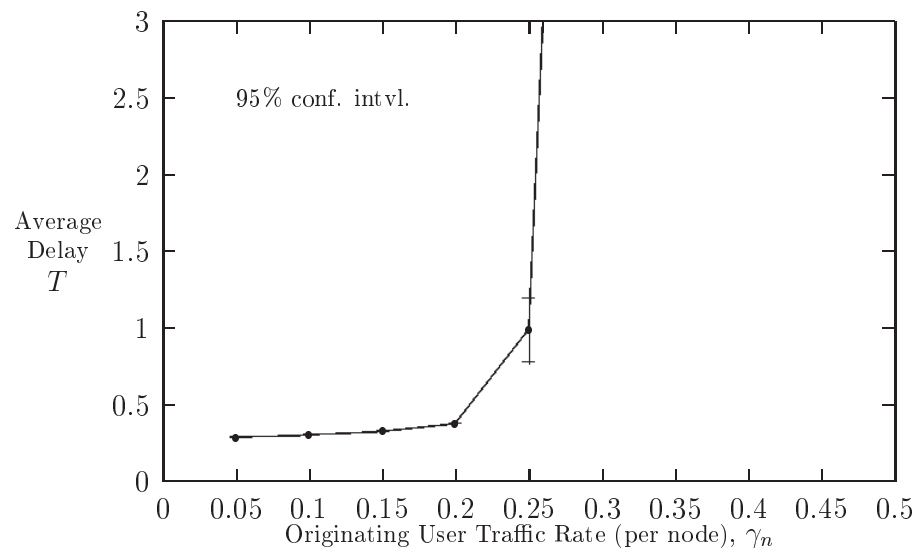


Figure 7.10: Flooding performance in the mobile radio network model

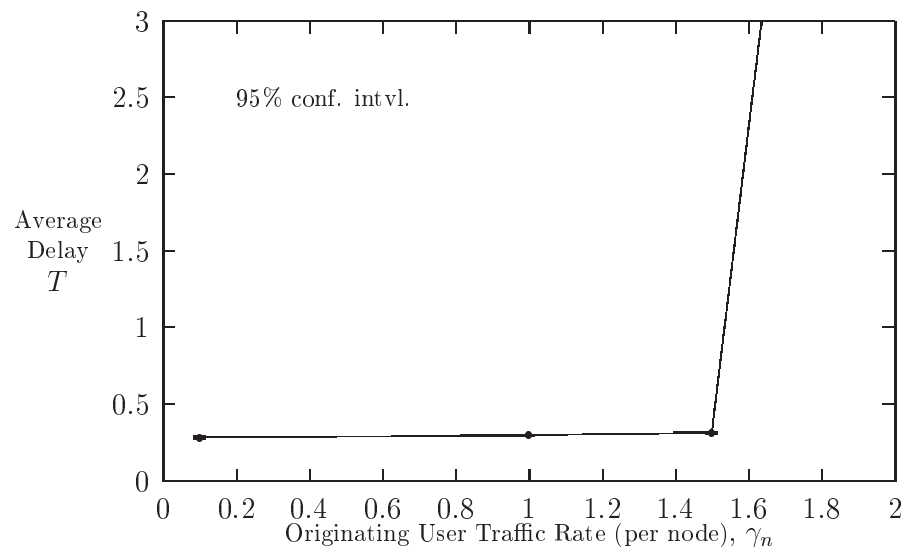


Figure 7.11: Hybrid routing performance in the mobile radio network model

Table 7.1: Routing power performance in the mobile radio network model

Algorithm	γ'_n	p_n	T	S/T
Minimum Hop	—	—	—	—
Flooding	0.2	0.99	0.38	0.53
Hybrid	1.5	0.99	0.31	4.79

dure operate as expected under the now modelled dynamic conditions, but its maximum operating point is almost an order of magnitude beyond that of constrained flooding. This represents a significant performance improvement over constrained flooding while minimum hop routing is inoperable under the same conditions, compare routing power values in Table 7.1. It is difficult to directly compare these results for hybrid routing with those of the 64 node, degree 4 Manhattan network model. Intensity of change, degree of topology change and the distribution of traffic loads are difficult to estimate in the mobile radio network dynamic topology change model.

Chapter 8

Conclusions

Routing procedures typically employed in quasi-static and dynamic networks have been investigated by examining the various factors contributing to routing overheads when operating in networks that are subjected to varying degrees of dynamic behaviour. Of particular interest was the impact of these routing overheads on overall network delay and throughput performance. Instead of considering routing algorithm performance from a single perspective such as execution speed or communication overhead, the routing procedures' performance have been measured as the overall cost or overhead under dynamic operational conditions. This more accurately identifies the true causes of network performance degradation. Within this framework it has been possible to envisage schemes whereby routing protocols are either switched or combined to counter these inherent routing costs and lead to a more efficient routing strategy. Following from this, a hybrid routing procedure has been proposed and verified using simulation modelling to reduce routing costs in networks subjected to static, quasi-static and dynamic operational conditions. Significant improvements in overall performance resulting from using this hybrid routing strategy have been extensively demonstrated.

8.1 Identifying Dominant Routing Costs

In Chapter 2 routing costs associated with general flooding and shortest path algorithms were examined. Flooding incurs high routing costs due to packet duplication but is independent of network topology, which makes it robust under dynamic net-

work conditions. Routing table update overheads are significant for shortest path routing while the forwarding of actual user traffic is very efficient. Comparison of the two observations led to the concept of switching routing protocols in the network depending on whether operational conditions are relatively static or very dynamic. Unfortunately this approach was predicted (and later confirmed in Chapter 5) to be very difficult to implement in a decentralised manner.

Modelling execution of the minimum hop (shortest path) routing procedure within a simulated network revealed the dominating affect of routing table uncertainty on overall network performance. This was indicated by the high occurrence of end-to-end retransmissions resulting from lost user packets when operating the network under dynamic conditions. Chapter 6 reinforced this observation where the next best path strategy effectively traded routing table uncertainty for increased delay. By comparing routing costs between Figures 5.7 and 6.1 when $\mathcal{I} = 0.04$, it can be concluded that the update costs and retransmission costs caused by uncertainty in Figure 5.7 are significantly greater than the update costs and longer transit paths present in Figure 6.1.

High levels of routing table uncertainty reflect the inability of shortest path routing procedures to track dynamic changes occurring within a network. This characteristic is expected because algorithms of this type are generally designed to operate only in quasi-static networks. While convergence performance alone provides an indication of an algorithms ability to track change, the relative dominance of convergence speed over update message overheads under dynamic conditions is not evident from simple observation of typical algorithm performance metrics (such as those listed in Table 2.1). However, by appreciating the complex interactions of these issues within a network, it has been possible to anticipate the general performance gains achievable by combining routing strategies.

8.2 A Solution to Routing Table Uncertainty

Multiple protocol switching was judged unattractive because of the requirement for coordination to form regions or domains within the network and establish interfaces between them. This would be difficult in a distributed network architecture. Fur-

thermore, this approach tended to address the more general issue of operating either flooding or shortest path routing depending on whether the network is dynamic or quasi-static, without focussing on the specific causes of high routing overheads. The hybrid routing procedure developed using local broadcasting directly counters the dominating effect of routing table uncertainty using a much finer level of local control. This has led to the superior overall performance demonstrated in Chapter 5.

Of particular interest is the ability of hybrid routing to continuously self adapt to changing network conditions both spatially and with time using a totally distributed approach. This robust capability, demonstrated in Chapter 7, provides a survivable routing solution in addition to improving network utilisation. Moreover, by basing the technique on deterministic algorithms, confidence in operation under previously unproven situations is increased.

From results in Chapter 5, hybrid routing became less efficient relative to constrained flooding as network size increased (based on the dynamic behaviour model used). In general consideration of this result, it can be concluded that in very large and potentially dynamic networks some form of hierarchical structure should be overlayed onto smaller nonhierarchical clusters of nodes executing hybrid routing. This approach of interconnecting network domains is typical of networks used in practice. However, under more realistic situations where dynamic behaviour occurs regionally and varies with time, hybrid routing performance was significantly superior to minimum hop routing or flooding alone (see Figure 7.7). This result is expected because localised flooding is only invoked in unstable regions of the network while more efficient shortest path routing occurs elsewhere. Moreover, with this technique there is no need for any coordination if an unstable region moves or varies in intensity.

8.3 Recommendations for Further Study

A natural extension of this study would be to investigate the use of a more generalised least cost (rather than minimum hop) shortest path algorithm as the basis for hybrid routing. A least cost routing procedure could be designed to adapt to traffic load conditions and variations in link quality, such as bit error rate. While

the Jaffe-Moss algorithm was implemented for minimum hop routing in this thesis, it is capable of efficiently supporting least cost routing. The focus on network connectivity within this thesis enabled a good understanding of the key issues. However, uneven traffic distribution has clearly degraded network performance causing reduced throughput before the onset of saturation load.

It is not intuitively obvious what performance gains are possible using least cost routing. Under a given set of dynamic conditions, least cost routing would adapt to both link failure and changes in link cost caused by traffic load variations. This will certainly increase the amount of routing table update traffic while at the same time reduce network congestion. Resultant network performance will be dependent upon the link cost function and how frequently it triggers routing table updates. Also, if link cost increases do not reach the value ∞ , user traffic does not need to be discarded reducing the frequency and associated load of local broadcasting if hybrid routing is based on least cost algorithms. Hybrid routing and the simulation models used within this study can readily be adapted to provide insight into the performance of least cost routing.

When developing hybrid routing, the Jaffe-Moss minimum cost algorithm was selected because it is a relatively high performance algorithm in terms of convergence speed, low update traffic load and because routing paths are constructed loop free. In addition, its technique of indicating unknown successor nodes within routing tables suited the implementation of hybrid routing. However, this does not imply that the Jaffe-Moss algorithm is the ideal routing procedure for hybrid routing. Other shortest path algorithms with superior performance may also readily enable detection of routing table uncertainty and lead to a more efficient hybrid routing strategy. A similar argument can be applied to the selection of constrained flooding for use in hybrid routing. Perhaps incorporating a mechanism restricting the flooding process (such as controlled flooding's assignment of packet wealth) could lead to improved overall performance.

From the investigations reported in this thesis, we see that hybrid routing offers potential as an efficient routing strategy for networks subjected to varying degrees of dynamic behaviour. Close examination of practical networks that could benefit from the application of hybrid routing requires consideration and subsequent perfor-

mance evaluation. Some interesting candidates for further examination may include terrestrial mobile packet radio networks, personal mobility services within fixed or deployed networks and perhaps multi-satellite networks. A model based on a practical packet radio network would require greater emphasis on accuracy of modelling the true dynamic behaviour. More accurate traffic models would also be needed.

Mobility management in personal communications is similar in its role to topology control as defined within this thesis. So in support of personal mobility services, a routing strategy is also required to establish appropriate paths for individual source destination connections. If a decentralised architecture is required, perhaps for reasons of survivability and processing efficiency, hybrid routing could offer attractive advantages if excessive mobility is encountered.

As mentioned above, another practical application of hybrid routing could be for multi-satellite networks, such as large constellations of Low Earth Orbit satellites. Networks such as these will inevitably exhibit dynamic operational characteristics. Topology control and routing functions will again be required and with the current trend towards the use of lower cost, less reliable space segments; distributed architectures may well be required to improve the reliability. Although such satellite networks will be very dynamic, there exists a high level of predictable behaviour within satellite orbits enabling individual nodes to predict upcoming changes to intersatellite topology connectivity. This characteristic could be exploited to reduce the apparent level of dynamic behaviour [118], but perhaps on closer investigation this may only provide assistance to topology control, rather than routing.

Bibliography

- [1] J. Bellamy, *Digital Telephony*. John Wiley and Sons, second ed., 1990.
- [2] A. Orda, R. Rom, and M. Sidi, “Minimum delay routing in stochastic networks,” *IEEE ACM Transactions on Networking*, vol. 1, no. 2, pp. 187–198, April 1993.
- [3] A. Tanenbaum, *Computer Networks*. Prentice-Hall International, second ed., 1989.
- [4] M. Schwartz, *Telecommunication Networks: Protocols, Modeling and Analysis*. Addison-Wesley, 1988.
- [5] M. Schwartz and T. Stern, “Routing techniques used in computer communication networks,” *IEEE Transactions on Communications*, vol. COM-28, no. 4, pp. 539–552, April 1980.
- [6] W. Hsieh and I. Gitman, “Routing strategies in computer networks,” *IEEE Computer Magazine*, pp. 46–56, June 1984.
- [7] J. McQuillan, I. Richer, and E. Rosen, “The new routing algorithm for the ARPANET,” *IEEE Transactions on Communications*, vol. COM-28, no. 5, pp. 711–719, May 1980.
- [8] J. Hayes, *Modelling and Analysis of Computer Communications Networks*. Plenum Press, N.Y., 1984.
- [9] U. Pooch, D. Machuel, and J. McCahn, *Telecommunications and Networking*. CRC Press, 1991.
- [10] M. Schwartz, *Computer Communication Network Design and Analysis*. Prentice-Hall, Englewood Cliffs, N.J., 1977.
- [11] G. Fultz and L. Kleinrock, “Adaptive routing techniques for store-and-forward computer communications networks,” in *IEEE International Conference on Communications proceedings*, pp. 39.1–39.8, June 1971.

- [12] T.-S. Yum, "The design and analysis of a semidynamic deterministic routing rule," *IEEE Transactions on Communications*, vol. COM-29, no. 4, pp. 498–504, April 1981.
- [13] H. Rudin, "On routing and 'delta-routing': A taxonomy and performance comparison of techniques for packet-switched networks," *IEEE Transactions on Communications*, vol. COM-24, no. 1, pp. 43–59, January 1976.
- [14] P. Chemouil, J. Filipiak, and P. Gauthier, "Performance issues in the design of dynamically controlled circuit-switched networks," *IEEE Communications Magazine*, pp. 90–95, October 1990.
- [15] G. Gopal, C.-K. Kim, and A. Weinrib, "Dynamic network configuration management," in *IEEE International Conference on Communications proceedings*, vol. 2, pp. 302.2.1–302.2.7, 1990.
- [16] R. Garzia and M. Garzia, *Network Modeling, Simulation, and Analysis*. Marcel Dekker, Inc., 1990.
- [17] G. Ash, "Design and control of networks with dynamic nonhierarchical routing," *IEEE Communications Magazine*, pp. 34–40, October 1990.
- [18] R. Ashany, D. Ferrari, and J. Pasquale, "Application of AI techniques to adaptive routing in wide-area networks," in *IEEE Seventh Annual International Phoenix Conference on Computers and Communications proceedings, Scottsdale, AZ*, pp. 157–160, March 1988.
- [19] K. Narendra and M. Thathachar, *Learning Automata An Introduction*. Prentice Hall, Englewood Cliffs, N.J., 1989.
- [20] S. Pithani and A. Sethi, "A fuzzy set delay representation for computer network routing algorithms," in *Second International Symposium on Uncertainty Modeling and Analysis proceedings*, pp. 286–293, April 1993.
- [21] D. Maravall, "Probabilistic and fuzzy learning automata for the optimal allocation of resources in a distributed system," *Advances in Modelling & Analysis B*, vol. 30, no. 3, pp. 31–34, 1994.
- [22] P. Mars, "Optimising PSS using learning automata," in *IEE Colloquium on "Advances in Optimisation", (Digest No. 45), London*, pp. 3/1–3/3, March 1989.

- [23] A. Vasilakos, "On the behaviour of learning algorithms in a changing environment with application to data network routing problem: Models, convergence analysis and optimality," in *Computer Standards & Interfaces 11*, pp. 29–39, Elsevier Science Publishers B.V. (North-Holland), 1990.
- [24] K. Narendra and M. Thathachar, "On the behaviour of a learning automaton in a changing environment with application to telephone traffic routing," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-10, no. 5, pp. 262–269, May 1980.
- [25] K. Narendra, E. Wright, and L. Mason, "Application of learning automata to telephone traffic routing and control," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-7, no. 11, pp. 785–792, November 1977.
- [26] A. Economides, P. Ioannou, and J. Silvester, "Decentralized adaptive routing for virtual circuit networks using stochastic learning automata," in *IEEE INFOCOM proceedings*, pp. 6C.4.1–6C.4.10, 1988.
- [27] A. Economides and J. Silvester, "Optimal routing in a network with unreliable links," in *IEEE Computer Networking Symposium proceedings, Washington*, pp. 288–297, April 1988.
- [28] B. Daneshrad and S. Morgera, "Application of stochastic automaton theory for routing in a packet-switched network," in *IEEE MILCOM proceedings*, pp. 11.1.1–11.1.5, 1989.
- [29] R. Gibbens, F. Kelly, and S. Turner, "Dynamic routing in multiparented networks," *IEEE ACM Transactions on Networking*, vol. 1, no. 2, pp. 261–270, April 1993.
- [30] G. Ash, "Real-time network routing in a dynamic class-of-service network," in *TELETRAFFIC AND DATATRAFFIC in a Period of Change, ITC-13* (A. Jensen and V. Iversen, eds.), pp. 187–194, Elsevier Science Publishers B.V. (North-Holland), 1991.
- [31] M. Meketon and D. Topkis, "Adaptive routing for damaged networks," in *IEEE MILCOM proceedings*, pp. 282–288, 1983.
- [32] B. Davies and T. Davies, "The application of packet switching techniques to combat net radio," *Proceedings of the IEEE*, vol. 75, no. 1, pp. 43–55, January 1987.

- [33] J. Garcia-Luna-Aceves, "Analysis of routing strategies for packet radio networks," in *IEEE INFOCOM proceedings*, pp. 292–302, 1985.
- [34] J. Hagouel, *Issues in Routing for Large and Dynamic Networks*. PhD thesis, Columbia University, New York, 1983.
- [35] W. Tajibnapis, "A correctness proof of a topology information maintenance protocol for distributed computer networks," *Communications of the ACM*, vol. 20, no. 7, pp. 477–485, July 1977.
- [36] D. Tipper and M. Sundareshan, "An optimal control approach to decentralized dynamic virtual circuit routing in computer networks," in *IEEE INFOCOM proceedings*, pp. 926–933, 1990.
- [37] J. Burgin, "Routing and resource control in the broadband ISDN," *Australian Telecommunication Research*, vol. 22, no. 1, pp. 3–19, 1988.
- [38] R. Onvural, *Asynchronous Transfer Mode Networks Performance Issues*. Artech House, 1993.
- [39] N. Deo and C.-Y. Pang, "Shortest-path algorithms: Taxonomy and annotation," *Networks*, vol. 14, pp. 275–323, 1984.
- [40] W.-K. Chen, *Theory Of Nets: Flows In Networks*. John Wiley and Sons, 1990.
- [41] B. Waxman, "Routing of multipoint connections," *IEEE Journal on Selected Areas In Communications*, vol. 6, no. 9, pp. 1617–1622, December 1988.
- [42] K. Cooke and E. Halsey, "The shortest route through a network with time dependent internodal transit times," *Journal of Mathematical Analysis and Applications*, vol. 14, pp. 493–498, 1966.
- [43] E. Klafszky, "Determination of shortest path in a network with time-dependent edge-lengths," *Math. Operational Research and Statistics*, pp. 255–257, 1972.
- [44] B. Gavish and I. Neuman, "Routing in a network with unreliable components," *IEEE Transactions on Communications*, vol. 40, no. 7, pp. 1248–1258, July 1992.
- [45] L. Ford and D. Fulkerson, *Flows in Networks*. Princeton University Press, 1962.
- [46] R. Bellman, "On a routing problem," *Quarterly Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

- [47] J. McQuillan, G. Falk, and I. Richer, "A review of the development and performance of the ARPANET routing algorithm," *IEEE Transactions on Communications*, vol. COM-26, no. 12, pp. 1802–1811, December 1978.
- [48] R. Gallager, "A minimum delay routing algorithm using distributed computation," *IEEE Transactions on Communications*, vol. COM-25, no. 1, pp. 73–85, January 1977.
- [49] P. Merlin and A. Segall, "A fail safe distributed routing protocol," *IEEE Transactions on Communications*, vol. COM-27, no. 9, pp. 1280–1287, September 1979.
- [50] A. Segall, "Distributed network protocols," *IEEE Transactions on Information Theory*, vol. IT-29, no. 1, pp. 23–35, January 1983.
- [51] S. Soloway and P. Humblet, "Distributed network protocols for changing topologies: A counter example," *IEEE Transactions on Communications*, vol. COM-39, no. 3, pp. 360–361, March 1991.
- [52] J. Jaffe and F. Moss, "A responsive distributed routing algorithm for computer networks," *IEEE Transactions on Communications*, vol. COM-30, no. 7, pp. 1758–1762, July 1982.
- [53] J. Gross and R. Ziemer, "Distributed routing algorithm performance in hostile environments," in *IEEE MILCOM proceedings*, pp. 11.5.1–11.5.5, 1989.
- [54] J. Garcia-Luna-Aceves, "A new minimum-hop routing algorithm," in *IEEE INFOCOM proceedings*, pp. 170–180, 1987.
- [55] C. Hedrick, "Routing Information Protocol, Request For Comments 1058," *Internet Engineering Task Force*, June 1988.
- [56] G. Malkin, "RIP version 2, carrying additional information, Request For Comments 1723," *Internet Engineering Task Force*, November 1994.
- [57] C. Huitema, *Routing in the Internet*. Prentice Hall, Englewood Cliffs, NJ, 1995.
- [58] J. Moy, "OSPF version 2, Request For Comments 1583," *Internet Engineering Task Force*, March 1994.
- [59] M. Borden, E. Crawley, B. Davis, and S. Batsell, "Integration of real-time services in an IP-ATM network architecture, Request For Comments 1821," *Internet Engineering Task Force*, August 1995.

- [60] C. White, E. Sykes, and J. Morrow, "An analytical approach to the dynamic topology problem," *Telecommunication Systems*, vol. 3, no. 3 and 4, pp. 397–413, March 1995.
- [61] T. Vu and S. Risner, "Performance and cost of broadcast routing algorithms in the strategic defense system terrestrial network," in *IEEE MILCOM proceedings*, pp. 29.2.1–29.2.5, 1991.
- [62] O. Lesser and R. Rom, "Routing by controlled flooding in communication networks," in *IEEE INFOCOM proceedings*, pp. 910–917, 1990.
- [63] Y. Park, C. Un, and S. Lee, "Performance analysis of flood search algorithm with MFDL path in circuit switched networks," in *IEEE MILCOM proceedings*, pp. 21.5.1–21.5.5, 1992.
- [64] D. Hübner, J. Kassubek, and F. Reichert, "A distributed multihop protocol for mobile stations to contact a stationary infrastructure," in *Third IEE Conference on Telecommunications proceedings, Conference Publication Number 331*, pp. 204–207, 1991.
- [65] Y. Lin and J. Yee, "A distributed routing algorithm for virtual circuit data networks," in *IEEE INFOCOM proceedings*, pp. 200–207, 1989.
- [66] M. Frankel, "Survivable command, control and communication," in *IEEE MILCOM proceedings*, pp. 30.1.1–30.1.4, 1987.
- [67] E. Wong and T.-S. Yum, "Maximum free circuit routing in circuit-switched networks," in *IEEE INFOCOM proceedings*, pp. 934–937, 1990.
- [68] M. Schroeder and K. Newport, "Enhanced network survivability through balanced resource critically," in *IEEE MILCOM proceedings*, pp. 38.4.1–38.4.6, 1989.
- [69] F. Amer, Y.-N. Lien, and A. Ghieth, "Performance evaluation of a hierarchical hybrid adaptive routing algorithm for large scale computer communication networks," in *IEEE Computer Networking Symposium proceedings*, pp. 266–275, 1988.
- [70] D. Dimitrijevic, B. Maglaris, and R. Boorstyn, "Routing in multidomain networks," *IEEE Transactions on Networking*, vol. 2, no. 3, pp. 252–262, June 1994.
- [71] J. Hahn and D. Stolle, "Packet radio network routing algorithms: A survey," *IEEE Communications Magazine*, vol. 22, no. 11, pp. 41–47, November 1984.

- [72] M. Deierling, R. Iyer, and H. Erwin, "Dynamics of network performance," in *IEEE MILCOM proceedings*, pp. 29.3.1–29.3.4, 1991.
- [73] F. Hill, "Alternative routing for the CINC area communications system: What is required and how can it be done," in *IEEE MILCOM proceedings*, pp. 2.3.1–2.3.6, 1989.
- [74] E. Gafni and D. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Transactions on Communications*, vol. COM-29, no. 1, pp. 11–18, January 1981.
- [75] L. Gifford, "Adaptive routing and traffic control in damaged circuit switched networks," in *IEEE MILCOM proceedings*, pp. 1.2.1–1.2.6, 1987.
- [76] M. Johnson, "Analysis of routing table update activity after resource recovery in a distributed computer network," in *Seventeenth Annual Hawaii International Conference on System Sciences proceedings*, pp. 96–102, 1984.
- [77] M. Johnson, "Updating routing tables after resource failure in a distributed computer network," *Networks*, vol. 14, pp. 379–391, 1984.
- [78] J. Garcia-Luna-Aceves, "Loop free routing using diffusing computations," *IEEE ACM Transactions on Networking*, vol. 1, no. 1, pp. 130–141, February 1993.
- [79] D. Bertsekas and R. Gallager, *Data Networks*. Prentice Hall, 1987.
- [80] R. Boorstyn and A. Livne, "A technique for adaptive routing in networks," *IEEE Transactions on Communications*, vol. COM-29, no. 4, pp. 474–480, April 1981.
- [81] L. Kleinrock, *Queueing Systems Volume I: Theory*. John Wiley and Sons, 1975.
- [82] L. Kleinrock, *Queueing Systems Volume II: Computer Applications*. John Wiley and Sons, 1976.
- [83] L. Kleinrock, "Performance evaluation of distributed computer-communication systems," in *Queueing Theory and its Applications* (O. Boxma and R. Syski, eds.), pp. 1–57, Elsevier Science Publishers B.V. (North-Holland), 1988.
- [84] P. White, "The changing role of switching systems in the telecommunications network," *IEEE Communications Magazine*, vol. 31, no. 1, pp. 12–13, January 1993.

- [85] A. Burson and A. Baker, "Optimizing communications solutions," *IEEE Communications Magazine*, vol. 31, no. 1, pp. 15–19, January 1993.
- [86] S. Personick, "The evolving role of telecommunications switching," *IEEE Communications Magazine*, vol. 31, no. 1, pp. 20–24, January 1993.
- [87] C. Robinson, "Army's digital devices move presses tactical automation," *Signal*, pp. 25–32, November 1990.
- [88] M. Hewish, "Tactical area communications part 2: Non-European systems," *International Defense Review*, pp. 675–678, June 1990.
- [89] J. Rustad, R. Skaug, and A. Aasen, "New radio networks for tactical communication," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 5, pp. 713–727, June 1990.
- [90] R. Benjamin, "Special military features for communications networks," Technical Note STC TN-159, SHAPE Technical Centre, The Hague, September 1987.
- [91] J. Mathis and J. Garcia-Luna-Aceves, "Survivable multiband networking," in *IEEE MILCOM proceedings*, pp. 30.4.1–30.4.7, 1987.
- [92] W. Chou, A. Bragg, and A. Nilsson, "The need for adaptive routing in the chaotic and unbalanced traffic environment," *IEEE Transactions on Communications*, vol. COM-29, no. 4, pp. 481–490, April 1981.
- [93] G. Clapp, "Multiple protocol network management," in *The Technical Cooperation Program, Subgroup S, Technical Panel STP-8 workshop proceedings*, pp. A3d/1–A3d/10, September 1990.
- [94] V. Li and R.-F. Chang, "Proposed routing algorithms for the U.S. Army mobile subscriber equipment (MSE) network," in *IEEE MILCOM proceedings*, pp. 39.4.1–39.4.7, 1986.
- [95] L. Miller, R. French, J. Lee, and D. Torrieri, "MSE routing algorithm comparison," in *IEEE MILCOM proceedings*, pp. 2.2.1–2.2.5, 1989.
- [96] S. Corson and A. Ephremides, "A distributed routing algorithm for mobile radio networks," in *IEEE MILCOM proceedings*, pp. 11.2.1–11.2.4, 1989.

- [97] W. Thoet, A. Ephremides, and D. Baker, "The use of multiple protocols in multi-hop radio networks," in *IEEE International Conference on Communications proceedings*, vol. 2, pp. 268–273, 1990.
- [98] A. Segall, "The modeling of adaptive routing in data-communication networks," *IEEE Transactions on Communications*, vol. COM-25, no. 1, pp. 23–35, January 1977.
- [99] C. Brown and M. Schwartz, "Adaptive routing in centralized computer communication networks," in *IEEE International Conference on Communications proceedings*, pp. 47.12–47.16, June 1975.
- [100] W. Tsai, "Convergence of gradient projection routing methods in a asynchronous stochastic quasi-static virtual circuit network," *IEEE Transactions on Automatic Control*, vol. 34, no. 1, pp. 20–33, January 1989.
- [101] D. Tipper and M. Sundareshan, "Numerical methods for modeling computer networks under nonstationary conditions," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 9, pp. 1682–1695, December 1990.
- [102] D. Raychaudhuri and N. Wilson, "ATM-Based transport architecture for multiservices wireless personal communication networks," *IEEE Journal on Selected Areas in Communications*, vol. 12, no. 8, pp. 1401–1414, October 1994.
- [103] Z. Qin, M. Veeraraghavan, and T. L. Porta, "An improved mobile tracking and location procedure for ATM-based PCNs," in *IEEE GLOBECOM proceedings*, pp. 6–11, November 1995.
- [104] T. Robertazzi and A. Lazar, "Deflection strategies for the manhattan street network," in *IEEE International Conference on Communications proceedings*, pp. 51.3.1–51.3.7, 1991.
- [105] B. Davies, S. Swain, and K. Watkins, "Efficient implementation of X.25 on narrow-band packet radio systems," in *IEE Colloquium on "Military Communications - The Trend Towards Civil Standards"*, (Digest No. 55), pp. 6/1–6/5, April 1990.
- [106] P. Karn, H. Price, and R. Diersing, "Packet radio in the amateur service," *IEEE Journal on Selected Areas in Communications*, vol. 3, no. 3, pp. 431–439, May 1985.
- [107] R. Walpole and R. Myers, *Probability and Statistics for Engineers and Scientists*. Macmillan Publishing, N.Y., second ed., 1978.

- [108] H. Westman, ed., *Reference Data For Radio Engineers*. Howard W. Sams, N.Y., fifth ed., 1969.
- [109] F. Preparata and M. Shamos, *Computational Geometry, An Introduction*. Springer-Verlag, N.Y., 1985.
- [110] T. Midtbø, *Spatial Modelling by Delaunay Networks of Two and Three Dimensions*. PhD thesis, Norwegian Institute of Technology, University of Trondheim, 1993. Copy available through Internet at <http://www.iko.unit.no/tmp/term/term.html>.
- [111] D. Lee and A. Lin, “Generalized delaunay triangulation for planar graphs,” in *Discrete & Computational Geometry*, pp. 201–217, Springer-Verlag, N.Y., 1986.
- [112] L. Hu, “Topology control for multihop packet radio networks,” *IEEE Transactions on Communications*, vol. COM-41, no. 10, pp. 1474–1481, October 1993.
- [113] R. Sedgewick, *Algorithms in C*. Addison-Wesley, 1990.
- [114] L. Hu, “A novel topology control for multihop packet radio networks,” in *IEEE INFOCOM proceedings*, pp. 1084–1093, 1991.
- [115] R. Ziemer and R. Peterson, *Introduction to Digital Communication*. Macmillan Publishing, N.Y., 1992.
- [116] F. Stremler, *Introduction To Communication Systems*. Addison-Wesley, second ed., 1982.
- [117] R. Peterson, R. Ziemer, and D. Borth, *Introduction To Spread-Spectrum Communications*. Prentice Hall, Englewood Cliffs, N.J., 1995.
- [118] N. Shacham, “Protocols for multi-satellite networks,” in *IEEE MILCOM proceedings*, pp. 26.3.1–26.3.5, 1987.

Appendix A

Routing Power Calculations

This appendix contains tabulated data derived from simulation experiments to calculate routing power for comparison of the various algorithms' performance. Details on how these raw values have been obtained are discussed in Chapter 5, the following provides a brief summary.

Routing power is defined as average throughput divided by average delay. In this thesis, routing power is scaled on a per node basis because traffic load is evenly distributed to all network nodes and it more accurately reflects an individual user's average quality of service. Maximum operating point is determined from the delay versus originating user traffic load curves and indicates an average user experienced delay, T , corresponding to the maximum operating originating user traffic load γ'_n . Probability of successful transmission through the network from node n , p_n , can be obtained from simulation results giving the ratio of total packets received to packets sent for the same maximum operating user traffic load γ'_n . Using Equ. A.1, average throughput per node can be calculated by multiplying γ'_n with the corresponding probability of successful transmission through the network. Routing power can then be determined as S/T .

$$S = \frac{\sum_{n=1}^N \gamma'_n p_n}{N} \quad (\text{A.1})$$

Table A.1: Routing Power - 64 Nodes with Connectivity of Degree 4

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Flood	0.00	0.2	1.00	0.33	0.61
Flood	0.04	0.2	1.00	0.33	0.60
Flood	0.16	0.2	1.00	0.33	0.61
Minhop	0.00	8.0	1.00	0.43	18.54
Minhop	0.02	5.5	1.00	0.56	9.83
Minhop	0.04	4.0	0.99	0.90	4.40
Hybrid	0.00	8.0	1.00	0.43	18.54
Hybrid	0.04	5.0	1.00	0.46	10.93
Hybrid	0.08	3.5	1.00	0.46	7.59
Hybrid	0.16	1.5	1.00	0.49	3.07
Next	0.00	8.0	1.00	0.43	18.54
Next	0.04	7.0	1.00	0.65	10.74
Next	0.08	6.0	1.00	0.77	7.76
Next	0.16	4.0	0.99	0.92	4.32

Table A.2: Routing Power - 64 Nodes with Connectivity of Degree 8

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Flood	0.00	0.24	1.00	0.37	0.64
Flood	0.04	0.25	1.00	0.42	0.60
Flood	0.16	0.25	1.00	0.37	0.68
Minhop	0.00	17.00	1.00	0.30	56.80
Minhop	0.02	11.00	1.00	0.30	36.80
Hybrid	0.00	17.00	1.00	0.30	56.80
Hybrid	0.04	10.00	1.00	0.27	37.56
Hybrid	0.08	7.50	1.00	0.28	27.05
Hybrid	0.16	4.50	1.00	0.33	13.75

Table A.3: Routing Power - 128 Nodes with Connectivity of Degree 4

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Flood	0.00	0.11	1.00	0.50	0.22
Flood	0.04	0.12	1.00	0.57	0.21
Flood	0.16	0.12	1.00	0.56	0.21
Minhop	0.00	5.00	1.00	0.63	7.88
Minhop	0.02	1.75	0.99	0.84	2.08
Hybrid	0.00	5.00	1.00	0.63	7.88
Hybrid	0.04	2.00	1.00	0.66	3.03
Hybrid	0.08	0.75	1.00	0.69	1.08
Hybrid	0.16	0.20	0.99	1.19	0.17

Table A.4: Routing Power - Prioritised Routing Table Update Messages

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Minhop	0.00	8.0	1.00	0.43	18.54
Minhop	0.02	4.5	1.00	0.54	8.29
Minhop	0.04	2.0	1.00	0.51	3.88
Hybrid	0.00	8.0	1.00	0.43	18.54
Hybrid	0.04	3.0	1.00	0.32	9.34
Hybrid	0.08	2.5	1.00	0.33	7.61
Hybrid	0.16	1.5	1.00	0.32	4.73

Table A.5: Routing Power - 64 Nodes, 4 Degree Connectivity, $\mathcal{D} = 0.25$

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Flood	0.00	0.20	1.00	0.33	0.61
Flood	0.04	0.20	0.99	0.38	0.53
Flood	0.16	0.20	0.99	0.38	0.52
Minhop	0.00	8.00	1.00	0.43	18.54
Minhop	0.01	2.50	0.99	0.62	4.00
Minhop	0.02	2.00	0.99	0.65	3.04
Hybrid	0.00	8.00	1.00	0.43	18.54
Hybrid	0.02	2.00	0.99	0.45	4.41
Hybrid	0.04	1.50	0.99	0.51	2.92
Hybrid	0.08	1.50	0.99	0.77	1.92
Hybrid	0.16	0.75	0.98	1.11	0.66

Table A.6: Routing Power - 64 Nodes, 4 Degree Connectivity, Regional Change

Algorithm	$\mathcal{I}$	γ'_n	p_n	T	S/T
Flood	0.00	0.2	1.00	0.33	0.61
Flood	0.01	0.2	1.00	0.33	0.60
Flood	0.02	0.2	1.00	0.33	0.61
Flood	0.04	0.2	1.00	0.33	0.60
Flood	0.08	0.2	1.00	0.33	0.60
Minhop	0.00	8.0	1.00	0.43	18.54
Minhop	0.005	6.0	1.00	0.44	13.58
Minhop	0.01	4.0	1.00	0.41	9.80
Hybrid	0.00	8.0	1.00	0.43	18.54
Hybrid	0.01	5.5	1.00	0.40	13.81
Hybrid	0.02	5.0	1.00	0.42	11.88
Hybrid	0.04	4.0	1.00	0.44	9.08
Hybrid	0.08	3.0	1.00	0.53	5.63

Appendix B

Dynamic Scenario for the Mobile Packet Radio Network Model

Contained within this appendix are diagrams illustrating the dynamic topology scenario used in the simulation experiments for the mobile packet radio network modelled in Chapter 7. An overall representation of initial node positions and their subsequent trajectories followed during execution of the simulation can be obtained by superimposing Figures B.1 and B.2. Node trajectories are indicated by arrows whose magnitude represents distance travelled during 600 seconds of simulation runtime.¹ The two figures are illustrated separately to reduce the resulting clutter if all nodes were to be shown within a single diagram. Figure B.2 clearly highlights the correlation in node trajectories as an attempt to avoid pure stochastic dynamic behaviour in the model.

The last diagram in this appendix, Figure B.3, shows the Delaunay triangulation constructed from initial node positions by the central topology control function as described in Chapter 7. Note that this triangulation is depicted before any assessment of individual edge range or nodal degree has been undertaken.

¹In these figures, 1cm corresponds to 4km travelled by the nodes.

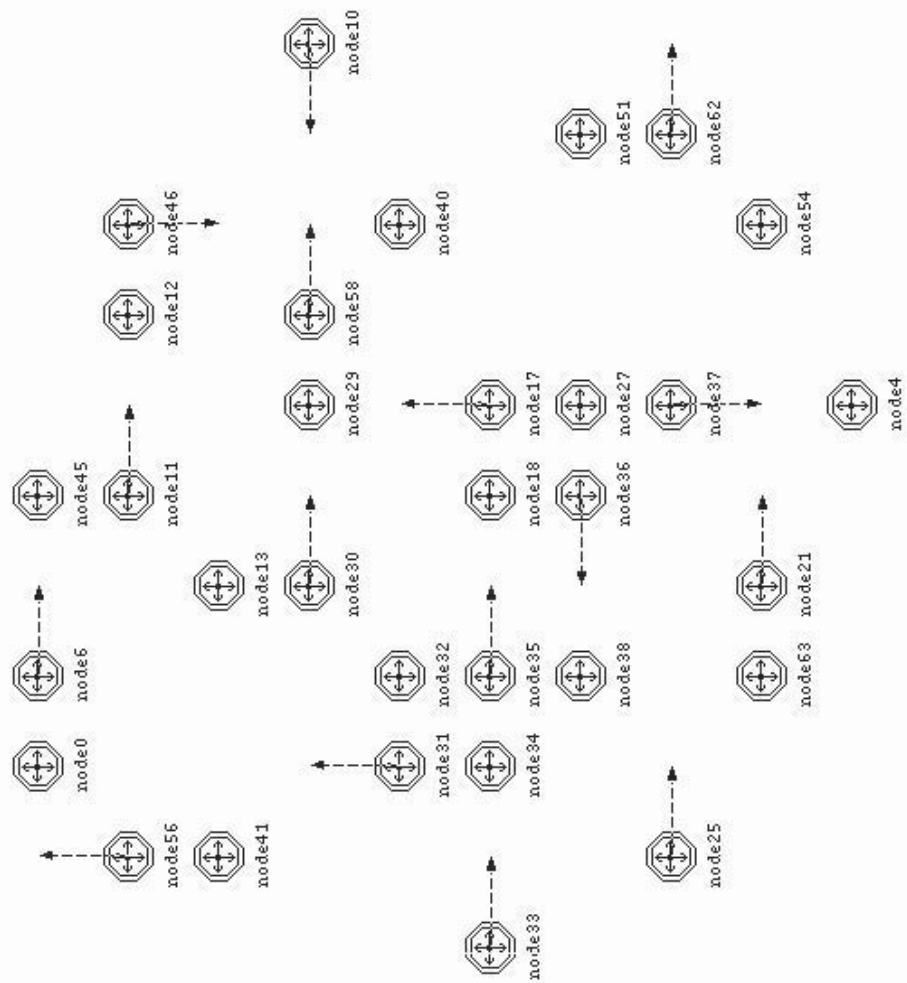


Figure B.1: Stationary nodes and nodes travelling at 30km/hr

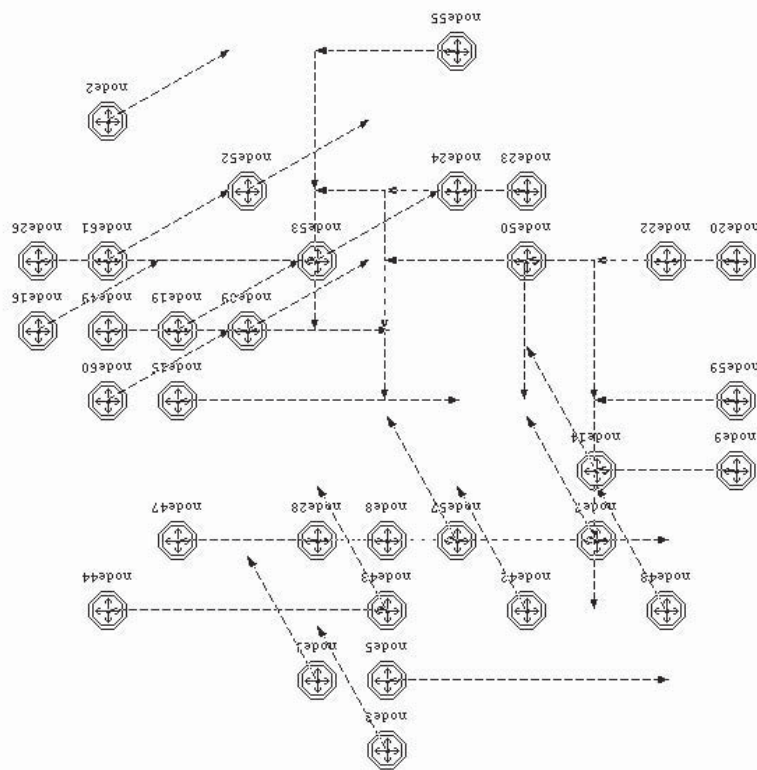


Figure B.2: Nodes travelling at 60km/hr and 120km/hr

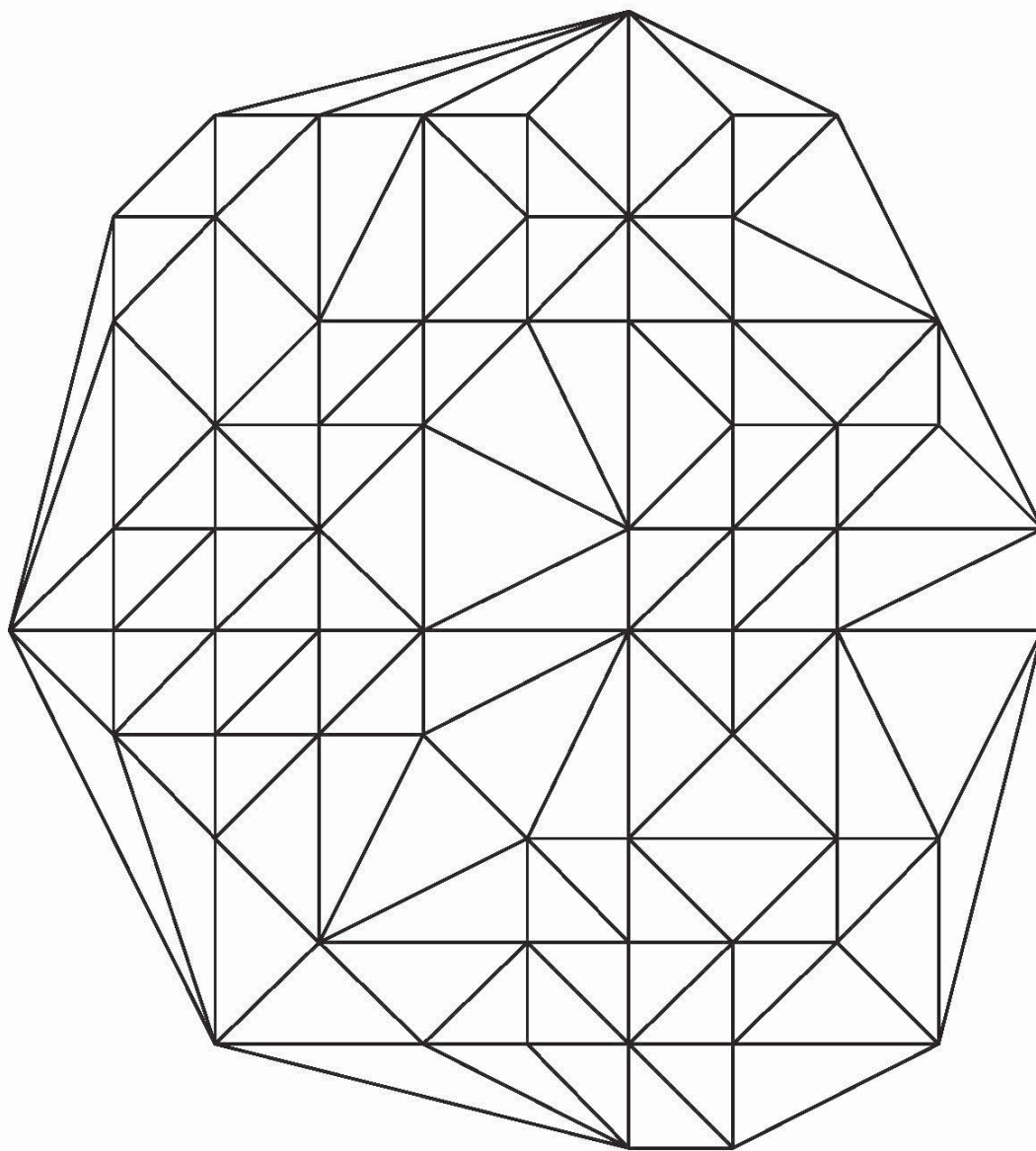


Figure B.3: Delaunay triangulation of initial node positions

Appendix C

Publications

This appendix contains a conference paper submitted for presentation at the IEEE International Conference on Communications to be held in Montreal, Canada, 8–12 June 1997.