1999

An analysis of delivery mechanisms for carrying IP traffic over ATM

Lorraine Michelle De Vere

University of Wollongong
An Analysis of Delivery Mechanisms for Carrying IP Traffic over ATM

A thesis submitted in fulfilment of the requirements for the award of the degree

Doctor of Philosophy

from

UNIVERSITY OF WOLLONGONG

by

Lorraine Michelle de Vere
Bachelor of Science Hons I (Computer Science)

DEPARTMENT OF ELECTRICAL, COMPUTER AND TELECOMMUNICATIONS ENGINEERING

1999
# Contents

Abbreviations ........................................... xiii

Abstract .................................................. xxii

Statement of Originality ............................. xxiii

Acknowledgements ................................. xxv

1 Introduction ........................................ 1
   1.1 Providing Internet Services over ATM ....... 1
   1.2 Thesis Outline .................................. 2
   1.3 Contributions .................................. 5
       1.3.1 Publications ............................... 7

2 The Future Provision of Internet Services .... 9
   2.1 Introduction ..................................... 9
   2.2 Categorisation of Applications Requiring Support . 10
   2.3 Internet Protocol Suite ......................... 11
       2.3.1 Unicast and Multicast Delivery Support .... 11
       2.3.2 Best Effort and Real Time Service Support ... 13
   2.4 ATM Protocol Suite ............................. 17
       2.4.1 Unicast and Multicast Delivery Support .... 18
       2.4.2 Best Effort and Real Time Service Support ... 22
   2.5 Proposals for Providing Internet Services over ATM . 23
   2.6 Key Issues ....................................... 26
       2.6.1 Address Resolution ......................... 26
       2.6.2 Delivery Mechanisms ....................... 28
       2.6.3 Providing the Required QoS ............... 28
   2.7 Conclusions ...................................... 31

3 Providing Internet Services over ATM ........ 33
   3.1 Introduction ..................................... 33
   3.2 Unicast Delivery Approaches and their Analysis ... 33
       3.2.1 Hop-by-Hop Approach ...................... 34
       3.2.2 Cut-Through Approach ...................... 35
       3.2.3 Comparison of Hop-by-Hop and Cut-Through Approaches . 36
3.2.4 Label Switching Approach .......................................................... 37
3.2.5 Comparison of Cut-Through, Label Switching and Hop-by-Hop Approaches .......................................................... 38

3.3 Intra-Subnet Multicast Delivery .............................................................. 41
  3.3.1 Alternative Intra-Subnet Multicast Delivery Approaches ............... 42
  3.3.2 Comparison of the VC Mesh and MCS Approaches .................... 45
  3.3.3 Performance Analysis of Intra-Subnet Approaches ..................... 46

3.4 Inter-Subnet Multicast Delivery .............................................................. 48
  3.4.1 Single Subnet Approach ............................................................. 48
  3.4.2 Hop-by-Hop Approach ................................................................. 51
  3.4.3 Label Switching Router Approach .......................................... 51
  3.4.4 NHRP Router Approach ............................................................. 52
  3.4.5 Performance Analysis Issues ....................................................... 55

3.5 The Multicast Delivery System ............................................................. 55

3.6 Conclusions ................................................................................................. 58
  3.6.1 Unicast Delivery .......................................................................... 58
  3.6.2 Multicast Delivery ....................................................................... 59

4 Methodology to Compare IP over ATM Approaches ................................ 63
  4.1 Introduction ................................................................................. 63
  4.2 Performance Measures ..................................................................... 64
    4.2.1 Unicast Performance Measures ........................................... 64
    4.2.2 Multicast Performance Measures ........................................... 65
  4.3 Analysis Requirements ..................................................................... 66
    4.3.1 Traffic Models ........................................................................ 66
    4.3.2 Segmentation and Reassembly ............................................. 68
    4.3.3 Flow Control ........................................................................... 71
    4.3.4 Multiple Flows ........................................................................ 73
    4.3.5 Accounting for Signalling Traffic ........................................... 73
    4.3.6 Accurately Modelling Routers and Switches ....................... 73
    4.3.7 Modelling Multicast Flows ................................................... 75
    4.3.8 Scalability ............................................................................... 76
  4.4 Analysis Methodology Description ..................................................... 76
    4.4.1 QNA Traffic Modelling ........................................................... 81
    4.4.2 Modelling Segmentation and Reassembly ................................ 81
    4.4.3 Calculating the Impact of Flow Control on Delay ................... 82
    4.4.4 Modelling Multiple Flows and Total Response Time ............. 84
    4.4.5 Incorporating Signalling Network Performance .................... 86
    4.4.6 Modelling Switch and Router Behaviour ................................ 86
    4.4.7 QNA Scalability ................................................................... 87
  4.5 Queuing Network Models .................................................................... 88
  4.6 Conclusions ............................................................................................... 90
5 Unicast Delivery .......................... 93
  5.1 Introduction .......................... 93
  5.2 Analysis Methodology ............. 94
    5.2.1 WWW Overview .................. 95
    5.2.2 WWW Traffic Model .......... 97
    5.2.3 Network Model .................. 99
    5.2.4 Further Analysis Assumptions .. 101
    5.2.5 Unicast Delivery Performance Measures ...... 101
  5.3 Performance Analysis .............. 104
    5.3.1 Effect of the Available Bit Rate .... 105
    5.3.2 Effect of the Data Network Utilisation . 111
    5.3.3 Effect of the Signalling Network Characteristics . 115
    5.3.4 Effect of the Number of Hops and Distance . 117
    5.3.5 Cut-Through Approach Resource Requirements .... 118
    5.3.6 Transmitting Smaller Volumes of Data .... 125
    5.3.7 Transmitting Connection-less Traffic .... 127
  5.4 Conclusions ......................... 129

6 Multicast Delivery ...................... 133
  6.1 Introduction ......................... 133
  6.2 Analysis Methodology .............. 134
    6.2.1 DIS Overview .................... 134
    6.2.2 DIS Traffic Model .............. 135
    6.2.3 Network Model .................... 137
    6.2.4 Multicast Delivery Approaches .... 143
  6.3 Response Time Performance Analysis .... 145
    6.3.1 Effect of the Data Network Utilisation .... 145
    6.3.2 Effect of the Data Network Link Bit Rate .... 155
    6.3.3 Effect of the Background Traffic Parameters .... 159
    6.3.4 Effect of Rendezvous Point Location .... 163
    6.3.5 Effect of the Number of Senders and Receivers .... 167
  6.4 Conclusions ......................... 172

7 Multicast Delivery Approach VC Requirements ...... 177
  7.1 Introduction ......................... 177
  7.2 Edge Device VC Requirements .... 178
  7.3 Core Network VC Requirements .... 181
  7.4 VC Analysis Methodology .......... 181
  7.5 Number of Active Links Required .. 184
  7.6 VCs Managed per Core Node ....... 190
  7.7 Subnet VC Requirements .......... 196
  7.8 Total VC Requirements .......... 199
  7.9 Conclusions ......................... 203
## Multicast Dynamic Group Analysis

8.1 Introduction ................................................................. 207
8.2 Approaches for Dynamic Multicast Group Support .......... 208
8.3 Dynamic Group Analysis Methodology .......................... 209
8.4 Performance Analysis ..................................................... 213
  8.4.1 Adding New Senders ............................................... 213
  8.4.2 Adding New Receivers ............................................ 219
  8.4.3 Sensitivity of Addition Delay to Data Message Size .... 222
  8.4.4 Signalling Demands of Approaches ......................... 224
8.5 Conclusions ................................................................. 232

## Conclusions

9.1 Overview ........................................................................ 237
9.2 Major Findings ............................................................... 237
  9.2.1 Unicast Delivery ....................................................... 237
  9.2.2 Multicast Delivery ................................................... 238
  9.2.3 Overall ................................................................. 241
9.3 Recommendations ........................................................ 244
9.4 Future Work ................................................................. 245
List of Figures

2.1 Internet Architecture ......................................................... 12
2.2 Internet Multicast Delivery .................................................. 13
2.3 Comparison of Source and Shared Multicast Forwarding Trees .... 14
2.4 RSVP Operation ..................................................................... 15
2.5 Heterogeneous Receiver Example ........................................... 16
2.6 Point-to-Point VC Creation .................................................. 18
2.7 ATM Unicast Data Delivery ................................................... 19
2.8 Point-to-Point and Point-to-Multipoint ATM Multicast Delivery
   Options .................................................................................. 20
2.9 Alternative Approaches for Adding a Receiver to a Pt-Mpt VC .... 21
2.10 Alternative Address Resolution Mechanisms ............................ 27

3.1 Alternative Inter-Subnet Unicast Topologies ............................ 34
3.2 Creation of a Cut-Through VC using NHRP ............................. 36
3.3 Creation of an End-End Switched Path using Label Switching ...... 39
3.4 Alternative Intra-Subnet Multicast Delivery Approaches .......... 43
3.5 Alternative Inter-Subnet Multicast Delivery Approaches .......... 49
3.6 Adding a new Receiver in the Label Switching Approach .......... 52
3.7 Direct VC Creation using PIM-SM and NHRP .......................... 53
3.8 VC Requirements when Reassembly occurs at Boundary Routers . 54

4.1 Effect of Arrival S.C.V. on Delay Performance ....................... 67
4.2 Effect of Segmentation and Reassembly on Delay Performance .... 69
4.3 The TCP Slow Start Mechanism ............................................ 71
4.4 Effect of TCP Slow Start on Application PDU Delay ................ 74
4.5 Comparison of Router and ATM Switch Performance ............... 75
4.6 Comparison of Pt-Pt and Pt-Mpt VC Performance .................... 77
4.7 Modelling Segmentation and Reassembly ............................... 82
4.8 Algorithm to Determine TCP Response Time .......................... 83
4.9 CPU Time to Solve Queuing Networks of Varying Size via QNA .. 87
4.10 Queuing Network Models .................................................... 89

5.1 Typical WWW Architecture ............................................... 95
5.2 WWW Message Flows ....................................................... 96
5.3 Mean WWW Response Time as Data Network Link Bit Rate Varies 106
5.4 Sensitivity of the Hybrid Approach to the Transmission Rate .... 108
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Number</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.5</td>
<td>WWW Response Time Variance as Data Network Link Bit Rate Varies</td>
<td>110</td>
</tr>
<tr>
<td>5.6</td>
<td>Mean WWW Response Time as Data Network Utilisation Varies</td>
<td>112</td>
</tr>
<tr>
<td>5.7</td>
<td>Mean WWW Response Time as the Number of Heavily Loaded Links Varies</td>
<td>114</td>
</tr>
<tr>
<td>5.8</td>
<td>Mean Response Time as Network Utilisation and Processor Speed Varies</td>
<td>116</td>
</tr>
<tr>
<td>5.9</td>
<td>Mean Response Time as the Distance Per Hop Varies Assuming 8 Hops</td>
<td>119</td>
</tr>
<tr>
<td>5.10</td>
<td>Cut-Through Approach Resource Requirements as the VC Holding Time Varies</td>
<td>121</td>
</tr>
<tr>
<td>5.11</td>
<td>Optimisation of Holding Time Considering Signalling and VC Resources</td>
<td>123</td>
</tr>
<tr>
<td>5.12</td>
<td>Sensitivity of the Optimal Holding Time to the Relative Signalling and VC Costs</td>
<td>124</td>
</tr>
<tr>
<td>5.13</td>
<td>Mean Response Time as the TCP Traffic Volume Varies (USA Server)</td>
<td>126</td>
</tr>
<tr>
<td>5.14</td>
<td>Mean Response Time as the UDP Traffic Volume Varies</td>
<td>128</td>
</tr>
<tr>
<td>6.1</td>
<td>JAMES Network Topology as of September 1996 [Con96]</td>
<td>138</td>
</tr>
<tr>
<td>6.2</td>
<td>Multicast Subnet Topology</td>
<td>139</td>
</tr>
<tr>
<td>6.3</td>
<td>DIS Host Locations on the James Network</td>
<td>141</td>
</tr>
<tr>
<td>6.4</td>
<td>Type I Message Mean Response Time as Data Network Utilisation Varies</td>
<td>147</td>
</tr>
<tr>
<td>6.5</td>
<td>Type I Message Response Time 95th Percentile as Data Network Utilisation Varies</td>
<td>148</td>
</tr>
<tr>
<td>6.6</td>
<td>Type I Message Response Time Jitter as Data Network Utilisation Varies</td>
<td>151</td>
</tr>
<tr>
<td>6.7</td>
<td>Type IV Message Mean Response Time as Data Network Utilisation Varies</td>
<td>153</td>
</tr>
<tr>
<td>6.8</td>
<td>Type IV Message Response Time Jitter as Data Network Utilisation Varies</td>
<td>154</td>
</tr>
<tr>
<td>6.9</td>
<td>Type I Message Response Time 95th Percentile as Data Link Bit Rate Varies</td>
<td>156</td>
</tr>
<tr>
<td>6.10</td>
<td>Type IV Message Mean Response Time as Data Link Bit Rate Varies</td>
<td>158</td>
</tr>
<tr>
<td>6.11</td>
<td>Type I Message Response Time 95th Percentile as Background Traffic Arrival S.C.V. Varies</td>
<td>160</td>
</tr>
<tr>
<td>6.12</td>
<td>Type I Message Response Time 95th Percentile as Background Traffic Message Size Varies</td>
<td>162</td>
</tr>
<tr>
<td>6.13</td>
<td>Type I Message Response Time 95th Percentile as RP Distance from Network Centre Varies</td>
<td>164</td>
</tr>
<tr>
<td>6.14</td>
<td>Type I Message Response Time Jitter as RP Distance from Network Centre Varies</td>
<td>165</td>
</tr>
<tr>
<td>6.15</td>
<td>Type I Message Response Time 95th Percentile as the Number of Interfaces to the RP Varies</td>
<td>166</td>
</tr>
<tr>
<td>6.16</td>
<td>Type I Message Response Time 95th Percentile as the Number of Hops between the RP and Network Centre Varies</td>
<td>168</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

6.17 Type I Message Response Time 95th Percentile as the Number of Senders Varies .......................................................... 169
6.18 Type I Message Response Time Jitter as the Number of Senders Varies ........................................................................ 171

7.1 Example Ten Node Generated Network ..................................... 182
7.2 Mean Number of Active Links Required as the Network Size Varies assuming 5 groups with 2 senders and 2 receivers ............. 185
7.3 Mean Number of VCs per Active Link assuming 5 groups with 2 senders and 2 receivers ....................................................... 186
7.4 Mean Number of VCs per Active Link as the Number of Groups Vary in 20 Node Networks ................................................. 188
7.5 Mean Number of VCs Managed per Core Node as Network Size Varies assuming 5 groups with 2 senders and 2 receivers .......... 191
7.6 Mean Number of VCs Managed per Core Node in Source Forwarding Trees as the Number of Groups Vary in 20 Node Networks .. 194
7.7 Mean Number of VCs managed per Core Node as the Number of Senders vary, assuming 20 node networks containing 5 groups each with 2 receivers - Source Forwarding Trees ............................. 195
7.8 Mean Number of VCs Initiated per Subnet, as the Number of Groups Varies assuming 2 senders and 2 receivers per group and 20 node networks ........................................................................ 197
7.9 Mean Number of VCs Terminated per Subnet as the Number of Groups Vary in 20 Node Networks with 2 Senders and 10 Receivers per Group .................................................................................. 200
7.10 Total VC Requirements Assuming Source Forwarding Trees and 5 groups each with 2 senders and 2 receivers ............................................ 201

8.1 Message Flows to Add New Senders ......................................... 211
8.2 Message Flows to Add New Receivers ....................................... 211
8.3 Mean Time for Data from a New Sender to reach all Receivers as the Sender Addition Rate Varies ........................................ 214
8.4 Mean Time for Data to reach a New Receiver as the Receiver Addition Rate Varies ......................................................... 220
8.5 Time for Data from a New Sender to reach Receivers as the Sender's Data Message Size Varies ............................................ 223
8.6 Signalling Processor Requirements When Adding Senders .... 226
8.7 Signalling Processor Requirements When Adding Receivers .... 230
# List of Tables

2.1 Classification of Proposals in Terms of Delivery and Service Class. 25

3.1 VC Requirements of Intra-Subnet Multicast Delivery Approaches . 43
3.2 Alternative Multicast Delivery Approaches .............................. 57

4.1 QNA Parameter Definitions ................................................. 80

5.1 HTTP Response and Request Size Statistics ......................... 97
5.2 HTTP Response Datagram Count Statistics .......................... 98
5.3 HTTP Response Datagram Size Statistics ......................... 98
5.4 Message Flow Arrival Statistics ....................................... 99
5.5 Message Size Statistics ................................................. 99
5.6 WWW Server Locations ................................................ 100
5.7 Queuing Analysis Assumptions ......................................... 101
5.8 WWW Scenario Hop and Distance Statistics ..................... 118

6.1 Number of DIS Entities and Their Location .......................... 135
6.2 DIS Message Flow Statistics ........................................... 137
6.3 Weather Message Datagram Flows ................................... 138
6.4 Multicast Transmission and Processing Delay Assumptions ...... 140

7.1 VC Requirements of Approaches ........................................ 179
7.2 Example of the Number of VCs Terminated by Receivers as the Number of Senders Varies ........................................ 180
7.3 Multicast Group Parameter Assumptions ............................. 183
7.4 Example Multicast Group Distribution ................................ 183

8.1 Multicast Delivery Approaches .......................................... 208
8.2 Number of New Receivers and Senders each Approach can Support 224
Abbreviations

ABR  Available Bit Rate
AAL  ATM Adaptation Layer
ACK  TCP Acknowledgment Packet
AREQUIPA  Application Requested IP over ATM
ARP  Address Resolution Protocol
ATM  Asynchronous Transfer Mode
bits/s  Bits per second
B-HLI  Broadband High Layer Information
B-ISDN  Broadband Integrated Services Digital Network
BUS  Broadcast and Unknown Server
CBR  Constant Bit Rate
CBT  Core Based Tree
CLIP  Controlled Load
CLIP  Classical IP over ATM
CLNAP  Connection-less Network Access Protocol
COIP  Conventional IP over ATM
CSCW  Computer Supported Collaborative Work
CSR  Cell Switching Router
cwnd  TCP Congestion Window
DNS  Domain Name System
DIS  Distributed Interactive Simulation
DoD  U.S. Department of Defense
Abbreviations

DVMRP  Distance Vector Multicast Routing Protocol
EARTH  EAsey IP multicast Routing THrough ATM clouds
EPD    Early Packet Discard
FCFS   First Come, First Served
FIFO   First In, First Out
FIH    Flow ID Handle
FIN    TCP Connection Termination Packet
FTP    File Transfer Protocol
GCID   Global Call Identifier
GS     Guaranteed Service
HTTP   Hypertext Transfer Protocol
IDMR   Inter Domain Multicast Routing
IETF   Internet Engineering Task Force
IFMP   Ipsilon Flow Management Protocol
IGMP   Internet Group Management Protocol
ILMI   Interim Local Management Interface
ION    Internetworking Over NBMA
IP     Internet Protocol
I-PNNI Integrated PNNI
IRA    Integrated Routing and Addressing
ISSLL  Integrated Services over Specific Lower Layers
ITU-T  International Telecommunication Union, Telecommunication Standardisation Sector
kbits/s Kilobits per second
LAN    Local Area Network
LANE   ATM Forum LAN Emulation Proposal
LEC    LAN Emulation Client
LECS   LAN Emulation Configuration Server
LES    LAN Emulation Server
LIH  Logical Interface Handle
LIJ  Leaf Initiated Join
LIS  Logical IP Subnet
LSR  Label Switching Router
mrouter Multicast Capable Router
MAC  Medium Access Control
MAN  Metropolitan Area Network
MARS Multicast Address Resolution Server
Mbits/s Megabits per second
MBone Multicast Internet Backbone
MCS  Multicast Server
MLIS Multicast Logical IP Subnet
MOSPF Multicast Extensions to Open Shortest Path First
MPOA  Multi Protocol over ATM
mpt-mpt Multipoint to Multipoint
MTU  Maximum Transmission Unit
NARP  NBMA Address Resolution Protocol
NBA  Network Based Address
NBMA  Non-Broadcast Multi-Access
NAS  NBMA Address Resolution Server
NHRP  NBMA Next Hop Resolution Protocol
NHS  Next Hop Server
NIC  Network Interface Card
NLIJ  Network Leaf Initiated Join
NNI  Network Node Interface
nrt-VBR Non-Real-Time Variable Bit Rate
OPWA  One Pass With Advertising
PDU  Protocol Data Unit
Abbreviations

PIM  Protocol Independent Multicast
PIM-DM  Protocol Independent Multicast - Dense Mode
PIM-SM  Protocol Independent Multicast - Sparse Mode
PNNI  Private Network Node Interface
pt-mpt  Point to Multipoint
pt-pt  Point to Point
PVC  Permanent Virtual Connection
QNA  Queuing Network Analyser
QoS  Quality of Service
rt-VBR  Real Time Variable Bit Rate
RCE  RSVP Co-ordination Entity
RFC  Request For Comments - IETF Document
RIJ  Root Initiated Join
RLIJ  Root Leaf Initiated Join
RP  Rendezvous Point
RSVP  Resource ReSerVation Protocol
SAR  Segmentation and Reassembly
s.c.v.  Squared Coefficient of Variation
SINP  Session Identity Notification Protocol
SS  Single Subnet
SVC  Switched Virtual Connection
SYN  TCP Connection Creation Packet
TCP  Transmission Control Protocol
TTL  Time To Live
TULIP  TCP and UDP over Lightweight IP
TUNIC  TCP and UDP over a Nonexistent IP Connection
UBR  Unspecified Bit Rate
UNI  User Network Interface
Abbreviations

VBR Variable Bit Rate
VC Virtual Channel Connection
VCI Virtual Connection Identifier
VENUS Very Extensive Non-Unicast Service
VP Virtual Path Connection
VPI Virtual Path Identifier
WAN Wide Area Network
WWW World Wide Web Internet Application
To God be the Glory
Abstract

In recent years, the variety, and use of Internet applications has grown enormously. As a result, the Internet is moving from providing solely unicast delivery and best effort service, to also providing multicast delivery and real time service support. Asynchronous Transfer Mode (ATM) is also being designed to support these two delivery and service classes. Furthermore, as the deployment of ATM in both the wide and local area has increased in recent years, there has been growing interest in the problem of providing Internet services over ATM networks.

Selecting a delivery mechanism for carrying Internet traffic over ATM infrastructure is a core issue that must be resolved. Many schemes for providing Internet services over ATM have been proposed, however there is no clear consensus in the literature of which scheme should be followed. Moreover, a detailed review of the literature reveals that little analysis examining the performance of these schemes has been performed. This thesis addresses this problem by presenting a detailed analysis of schemes for providing unicast and multicast Internet delivery over wide area ATM networks.

The first part of this thesis develops a queuing analysis framework to enable the detailed performance comparison of delivery schemes. This work extends existing queuing network methodology to model: point-to-multipoint ATM connections; the segmentation and reassembly of Internet packets into ATM cells; and the Internet Transmission Control Protocol Slow Start mechanism.

The second portion of this thesis develops a realistic application and network model for carrying World Wide Web traffic over the Internet. This model is used to compare the key alternative approaches for unicast delivery: the Hop-by-Hop approach; and the Buffered and Hybrid variants of the Cut-Through unicast delivery approach. We find that both the Buffered and Hybrid Cut-Through approaches provide significantly lower response times than the Hop-by-Hop approach, even when the end-to-end (or cut-through) VC must be created, as long as the signalling network is able to support the expected traffic volume. The analysis also shows that the Buffered approach out-performs the Hybrid approach in most network scenarios.

Furthermore, it was found that although the Cut-Through approach has higher VC and signalling resource requirements than the Hop-by-Hop approach, current switching equipment can easily support the resource requirements of a moderately sized World Wide Web proxy, regardless of the unicast delivery approach employed. Hence, when the signalling network resources are sufficient for the expected traffic volume, the Buffered Cut-Through approach should be used to carry
unicast Internet traffic over ATM networks.

The third portion of this thesis compares approaches for delivering multicast Internet traffic over ATM WANs. To provide a complete multicast delivery solution: intra and inter subnet, and forwarding tree delivery approaches are all required. The VC Mesh and MCS are the key alternative intra-subnet approaches. There are also several inter-subnet approaches which vary in whether they create direct ATM VCs across subnet boundaries, or place routers between subnets. We investigate the performance of the Hop-by-Hop, Boundary and Cut-Through inter-subnet approaches. Multicast delivery systems also vary in whether they utilise one Shared Forwarding Tree or multiple Source Forwarding Trees.

Data delivery, dynamic group membership performance, and resource requirements issues must be addressed when providing multicast delivery. This thesis examines data delivery performance measures including response time, jitter, and VC requirements. The performance of approaches when supporting dynamic multicast groups are also examined in terms of the time to add new senders or receivers, and the signalling network resources requirements.

We find that the Cut-Through and Boundary VC Mesh, Source Forwarding Tree approaches have the best response time performance. Furthermore, if the network is engineered such that potential bottleneck points in the MCS and Shared Forwarding Tree approaches have sufficient capacity, there is often no significant difference between the delay performance of the VC Mesh and MCS approaches, or between the Source and Shared Forwarding Tree approaches. However our analysis also clearly shows that the Source Forwarding Tree approach can have significantly higher resource requirements and sender and receiver addition delays than the Shared Forwarding Tree approach. This thesis also highlights that none of the multicast delivery approaches are able to meet the delay requirements of real-time applications. Hence resource reservation techniques and non-FIFO scheduling are critical when networks support real-time applications.

In summary, this thesis provides a detailed performance comparison of approaches for delivering unicast and multicast Internet traffic over wide area ATM networks. This is achieved via the development of an analytical framework and realistic network and traffic models. This thesis emphasises the trade-off between delivery approaches depending on whether delay performance or resource requirements are the greatest concern. Given this trade-off we see an important role for hybrid systems. This also falls in line with the multiple services platform principles of both Internet and ATM technologies. Internet applications, and users of these applications have a wide range of service requirements, and may be willing to pay different amounts for these services. This thesis clearly shows that the Hop-by-Hop approach is far superior to all other approaches in terms of resource requirements. Hence it should be employed as the basic service for carrying IP traffic over ATM networks. However, this thesis also shows that the Cut-Through and Boundary approaches can provide significant delay savings compared to the Hop-by-Hop approach. In most network scenarios, the Boundary approach has the delay characteristics of the Cut-Through approach, yet resource requirements which are significantly lower. Hence, we see a role for the Boundary approach, particularly for premium grade services.
Statement of Originality

This is to certify that the work described in this thesis is entirely my own, except where due reference is made in the text. No work in this thesis has been submitted for a degree to any other university or institution.

Signed

Lorraine de Vere
Acknowledgements

I would like to thank my thesis supervisors, Dr. Tony Eyers and Professor Gary Anido. I greatly appreciate their guidance and encouragement. I would also like to thank my colleagues at Lucent Technologies, Swindon, UK for their support while completing this thesis, particularly my manager Ioannis Kriaras. Finally I would like to thank my husband Chris Stacey, and my parents for all of their invaluable support.

Lorraine de Vere
Chapter 1

Introduction

1.1 Providing Internet Services over ATM

The subject of this thesis is the provision of Internet services over ATM networks. In the last few years the use of Internet services has grown enormously. Furthermore, the variety of applications being supported by the Internet is increasing. In response the Internet is moving from a best effort, unicast model, to an integrated services model, capable of supporting applications with a range of traffic characteristics and service requirements. Asynchronous Transfer Mode (ATM) has also been developed to provide a single integrated network capable of supporting applications with a range of characteristics and requirements. Furthermore, compared to many other network technologies, ATM has the additional advantage that, like the Internet, it can operate at a range of link rates over any distance or physical media. As a result, many researchers are currently investigating how to provide Internet services over ATM networks.

Many schemes for providing Internet services over ATM have been proposed. However a detailed review of the literature reveals that little analysis comparing these schemes has been performed, particularly when considering wide area ATM networks. Many agree that performance is a critical issue when providing Internet services over ATM [LE96, SST95], "the key issue when providing Internet services over ATM, is performance". A review of the literature also reveals that the selection of a delivery mechanism is central to the overall problem of providing Internet services over ATM. Hence this thesis provides a detailed analysis of schemes for providing unicast and multicast delivery over wide area ATM networks.
1.2 Thesis Outline

This thesis is divided into eight chapters. Chapter 2 describes the characteristics of applications that the Internet is evolving to support. An overview of the Internet and ATM protocol suites is then presented focusing on how they support applications requiring either unicast or multicast delivery and either best effort or real time service. This highlights the fact that Internet and ATM protocol suites meet application requirements very differently. As a result many proposals have appeared in the literature for providing Internet services over ATM.

The remainder of Chapter 2 describes the key issues that must be resolved to provide Internet services over ATM. This discussion reveals: (1) the central role delivery mechanisms play in the overall problem of providing Internet services over ATM; and (2) that there is still no clear consensus on which delivery approaches should be used to carry Internet traffic over ATM networks. Hence the rest of the thesis focuses on mechanisms for delivering unicast or multicast Internet traffic over ATM.

In Chapter 3 we review proposed approaches for delivering Internet traffic over ATM. Approaches are classified in terms of the delivery mechanism they provide: unicast, intra-subnet multicast, inter-subnet multicast, or multicast forwarding trees. Chapter 3 also reviews performance studies of Internet service provision over ATM. Most of these studies have focused on aspects of providing Internet services over ATM which are independent of the approach followed (e.g. connection management or the interaction between TCP and ATM congestion control). Other studies present empirical measurements of small ATM networks carrying Internet traffic. From our review of Internet traffic delivery over ATM research, we identify two major deficiencies: (1) Many different delivery mechanisms have been proposed, leading to much confusion, as to which approaches are best suited to which operating environments; and (2) This problem has been exacerbated by the lack of performance analysis of the delivery mechanisms. Chapter 3 identifies many performance issues that must be resolved before clear advice can be given to network operators to help them select an IP over ATM delivery mechanism. This thesis provides a detailed quantitative analysis of alternative delivery approaches that answers each of the identified performance issues.

Chapter 4 presents the methodology we have developed to analyse the performance of approaches for delivering Internet traffic over ATM. As stated above, without detailed performance comparisons to rely on, it is very difficult to decide which approach for delivering Internet traffic over ATM networks is the best for a given network environment. An important aspect of any analysis is deter-
1. Introduction

mining which performance measures are important when comparing approaches. Chapter 4 details the performance measures of greatest importance to users and network operators when comparing delivery approaches. This is followed by a description of the features an analysis methodology must exhibit to adequately study these performance measures. In particular the analysis requirements are created to resolve many of the short-comings in the small amount of performance analysis that has appeared in the literature to date. The remainder of the chapter presents the analysis methodology employed through-out the thesis, and describes how it meets each of the analysis requirements highlighted earlier in the chapter.

A performance analysis of alternative unicast delivery approaches is presented in Chapter 5. This addresses the shortcomings in the existing analysis of unicast delivery approaches detailed in Chapter 3. Due to the popularity of the World Wide Web, it is employed as the main application scenario through-out the chapter. However the performance of approaches are also considered when carrying other connection-oriented or connection-less application traffic. Approaches are compared in terms of their delay capabilities, and their signalling, VC and bit rate resource requirements. This analysis indicates that if the signalling network can support the expected volume of signalling traffic, the Cut-Through approach (where traffic is forwarded over end-end ATM VCs) produces significantly lower response times than the Hop-by-Hop approach (where traffic is reassembled and forwarded at intermediate IP routers), even if the direct VC must be created. Moreover, contrary to commonly held hypotheses, direct VCs should be created via the Buffered approach (where traffic is buffered while the VC is being created), not the Hybrid approach (where traffic is sent via the Hop-by-Hop approach until the end-end ATM VC is created). Finally, although the analysis showed the Cut-Through approach has significantly higher resource requirements than the Hop-by-Hop approach, these can be supported by current ATM network equipment.

In Chapter 6 the delay performance of alternative multicast delivery approaches are analysed. This analysis considers the entire multicast delivery system encompassing intra-subnet, inter-subnet and multicast forwarding tree approaches. Multicast applications have a variety of service requirements, ranging from best effort to strict delay guarantees. As a result this chapter assumes a Distributed Interactive Simulation (DIS) scenario involving several traffic flows each with different characteristics and QoS requirements. This means that the analysis extends to a wide variety of multicast applications. This analysis found that the Cut-Through Source Forwarding Tree approach produced the best delay performance of all approaches in all network scenarios. However, in many scenarios the performance of
the Cut-Through Source and Shared Forwarding approaches and the Boundary VC Mesh Source and Shared Forwarding approaches were similar. Furthermore, none of the delivery approaches were able to meet the QoS requirements of real-time traffic flows in all scenarios. Hence QoS provision mechanisms such as Integrated and Differentiated services must be employed, particularly if the network is highly utilised.

Clearly there is a performance trade-off between the delay performance of multicast delivery approaches and their resource requirements. Chapter 7 studies the VC requirements of alternative multicast delivery approaches. Analysis in the literature has considered the VC requirements of edge devices only. In this chapter the VC demands placed on subnet and core network switches and routers are examined. The performance of approaches, particularly in terms of switch and router VC requirements, will be sensitive to the network topology and the characteristics of the multicast groups being supported. Therefore this chapter makes recommendations based on VC requirements analysis in a variety of realistic network topologies. This analysis indicates a hybrid approach should be employed within subnets. This hybrid approach employs the VC Mesh approach for groups that contain one or two local senders, and the MCS approach for groups with more local senders. We also found that in terms of VC requirements, the Hop-by-Hop approach is far superior to all other inter-subnet approaches, requiring at most one pt-pt VC per link regardless of the number of multicast groups. Hence it should be employed in all environments where possible (i.e. where it is able to meet the application delay constraints).

Chapters 6 and 7 assumed that multicast group membership is static. However in practice many multicast applications, including DIS, have dynamic multicast group membership, i.e. where senders and receivers can join or leave the group at any time. Hence it is critical that the performance of alternative multicast delivery systems are compared when supporting dynamic multicast groups. Chapter 8 compares approaches in terms of the time, and signalling network resources required to add or remove senders and receivers. This analysis shows that there is no significant difference between the intra-subnet approaches unless the signalling network is highly utilised. Moreover the analysis shows that the Boundary and Cut-Through approaches should only be employed in conjunction with a Shared Forwarding Tree. Source Forwarding Trees have significantly higher signalling resource requirements and produce higher addition delays. The Hop-by-Hop approach is the preferred inter-subnet multicast delivery approach for dynamic multicast groups. It has the lowest signalling resource requirements, and also the best addition delay performance in most network scenarios.
Chapter 9 summarises the main findings of this thesis. Areas where future research scope exists are also identified. In general terms the main finding of this thesis is that there is an important role for hybrid delivery approaches within wide area ATM networks. Approaches that aggregate traffic as much as possible should be employed to provide a basic service. However premium grade services should be based on direct VCs due to their superior delay characteristics. The remainder of this chapter highlights the key contributions of this thesis.

1.3 Contributions

This section lists the contributions contained in this thesis. The section where this work is first discussed is also indicated.

- Detailed review of proposals for delivering Internet traffic over ATM, highlighting the shortcomings in current research in this area (Chapter 3).

- Identified the analysis methodology capabilities needed to compare alternative approaches to providing Internet services over ATM, and created an analysis framework that provides these capabilities based on the Queuing Network Analyser (QNA) (Chapter 4).

- Extended QNA to model traffic flows more accurately. This included: (1) enabling customer creation and combination factors to vary for different message flows, allowing protocol data unit segmentation and reassembly to be modelled; (2) Modelling the TCP slow start mechanism; (3) modelling point-to-multipoint traffic flows, allowing multicast delivery approaches to be analysed; and (4) deriving the per class arrival process squared coefficient of variation calculation, to model the arrival process at each queue more accurately (Chapter 4).

- Developed a realistic application and network model of carrying World Wide Web (WWW) traffic over the Internet using actual World Wide Web traffic traces (Chapter 5).

- Compared the Hop-by-Hop and Cut-Through unicast delivery mechanisms for carrying WWW traffic over wide area ATM networks. This showed that for current WAN bit rates the Cut-Through approach produces significantly lower delays than the Hop-by-Hop approach even when the Cut-Through approach must create a direct VC, regardless of the type of application traffic. Moreover this analysis showed that the Buffered approach for creating a
direct VC outperforms the Hybrid approach in the majority of performance scenarios (Chapter 5).

- Showed that a five minute VC holding time represents the best trade-off for a moderately sized proxy between: (1) the probability of needing to create a cut-through VC; (2) the number of active VCs; and (3) the frequency of VC creation requests for HTTP requests (assuming these performance measures are equally important). This analysis also enables network designers to select an appropriate VC holding time given the relative importance of these performance measures (Chapter 5).

- Regardless of the multicast delivery approaches employed we showed that it is not possible to meet the delay requirements of real time applications by forwarding all traffic in a best effort fashion. Hence QoS mechanisms such as Integrated or Differentiated Services must be employed (Chapter 6).

- Showed that the relative performance of intra-subnet multicast delivery approaches depends on the number of senders and receivers within the subnet. The VC Mesh approach should be employed for each group that has one or two local senders. The MCS approach should be employed for any active multicast groups that have three or more local senders. Furthermore, it is critical that the links to the MCS are engineered adequately, otherwise the MCS performance deteriorates significantly. (Chapters 6, 7 and 8).

- Similarly, we showed that there is no significant difference in the response times achieved by the Source and Shared Forwarding Tree approaches, if the RP is located close to the wire-line centre of the network (where traffic passes anyway), and the links to the RP and RP processor capacity are adequately engineered. However, when supporting dynamic multicast groups, the Shared Forwarding Tree approach can provide significantly better performance, both in terms of addition delay and resource requirements. (Chapters 6, 7 and 8).

- Through a detailed analysis of alternative inter-subnet multicast delivery approaches, we showed that the Hop-by-Hop inter-subnet multicast delivery approach produces significantly higher response times than the Cut-Through and Boundary approaches. However, the Hop-by-Hop approach has significantly lower VC and signalling requirements, and can produce significantly lower addition delays than both the Cut-Through and Boundary approaches. Hence the Hop-by-Hop approach should be employed for basic grade Internet services. For premium grade services, or other environments where delay
is a major concern, the Boundary approach should be employed. In many network scenarios this has similar delay performance to the Cut-Through approach, but significantly lower resource requirements. (Chapters 6, 7 and 8).

1.3.1 Publications


1. Introduction
Chapter 2

The Future Provision of Internet Services

2.1 Introduction

In recent years, the use of Internet applications has grown enormously. Furthermore, the variety of applications being supported by the Internet is increasing. Traditionally, the Internet supported text based applications including e-mail, telnet and the File Transfer Protocol (FTP). Now, many applications involve audio, video, images and text (e.g. conferencing, Computer Supported Collaborative Work (CSCW), the World Wide Web (WWW), and multi-player games). In addition, many of these applications have strict performance requirements (e.g. Internet telephony and multi-player games). In response, the Internet is moving from a best effort model, to an mixed services model, capable of supporting applications with a range of traffic characteristics and service requirements.

Asynchronous Transfer Mode (ATM), was also developed to provide a single integrated network capable of supporting applications with a range of traffic characteristics and service requirements. Hence, ATM is a leading contender, when considering which network topologies best support the mixed services Internet. Compared to many other network technologies, ATM has the additional advantage that, like the Internet, it can operate at a range of link rates over any distance and physical media. As a result, many researchers are currently investigating how to provide Internet services over ATM networks.

Although both the Internet and ATM protocol suites are designed to support all types of application, they do so very differently. The Internet follows a connection-less paradigm, where data is forwarded hop by hop towards the destination. Each Internet Protocol (IP) datagram contains sufficient addressing
information to allow it to be routed independently of all others. In contrast, ATM follows a connection oriented approach where a connection must be created before data can flow towards the destination. Furthermore, routing decisions are made as the connection is created. This means that ATM cells need to carry a connection identifier only, rather than addressing information. This difference in paradigms means that providing Internet services over ATM is a non-trivial problem which has many possible solutions.

This chapter reviews the current literature, highlighting the differences between the Internet and ATM protocol suite and how these differences must be resolved to provide Internet services over ATM. Section 2.2 describes the applications that both the Internet and ATM protocol suites must support. This is followed by an overview of the Internet and ATM protocol suites in Sections 2.3 and 2.4 respectively. Section 2.5 provides an overview of proposals for providing Internet services over ATM that have appeared in the literature, focusing on the types of application they support. Given, many proposals use the same general approaches to provide Internet services over ATM, the remainder of the thesis focuses on approaches rather than the specific details of individual proposals. Section 2.6 describes the key issues that must be resolved to provide Internet services over ATM. This discussion highlights the central role delivery mechanisms play in the overall problem of providing Internet services over ATM. Hence the remainder of this thesis focuses on alternative delivery mechanisms for carrying Internet traffic over ATM.

2.2 Categorisation of Applications Requiring Support

In the past the Internet only supported text based applications such as e-mail, telnet, and FTP. All of these applications were designed to operate in a best effort service environment. Hence, these applications accepted whatever level of service the Internet could currently provide. The Internet best effort service paradigm is one of the main reasons for its success. This is because it allows the Internet to operate over many types of link layer networks, including leased telephone lines, X.25, Ethernet and Token Ring networks.

A number of technological advances, particularly in the computer (e.g. increased processing power, better graphical displays), and telecommunications (e.g. increased link bit rates) industries has led to the development of a new range of applications. Rather than being solely text based, many of these applications incorporate audio, video, images and text (e.g. CSCW, WWW, multi-player games). Furthermore, many of these applications have strict performance requirements
2. The Future Provision of Internet Services

(e.g. Internet telephony). To support these real-time applications, the network must therefore provide some form of performance guarantee (e.g. a maximum end-end delay). In the remainder of this thesis these applications are referred to as real-time applications\(^1\), in contrast to best effort application that require no performance guarantees.

Traditional Internet applications also only involved a single sender and receiver, i.e., they required unicast delivery. In contrast, many new applications involve multiple senders and receivers, and hence require multicast delivery (e.g. conferencing, CSCW, Distributed Interactive Simulation (DIS), and multi-player games).

From the discussion above, it is clear that applications can be classified in terms of their service (best effort or real time) and delivery (unicast or multicast) requirements. Proposals for providing Internet services over ATM must support both classes of service and delivery. Sections 2.3 and 2.4 describe how the Internet and ATM protocol suites attempt to support each class of application respectively.

2.3 Internet Protocol Suite

The Internet comprises a large number of subnets connected by routers. The subnets operate a variety of link layer protocols, (e.g. Ethernet, Token Ring, ATM, or X.25). Hosts and routers attached to these subnets implement the Internet Protocol (IP), allowing data to be forwarded from senders to receivers hop by hop, as shown in Figure 2.1. A good introduction to the Internet protocol suite can be found in [Com91, Ste94].

2.3.1 Unicast and Multicast Delivery Support

The Internet provides unicast delivery by forwarding data from the sender, router to router, until the receiver's subnet is reached, as shown in Figure 2.1. The subnet link layer protocol is used to forward data within subnets (i.e. from the sender to router, router to router, and router to receiver).

In the late 1980's, the Internet community began to consider the development of a multicast capability. The Internet provides multicast delivery by creating forwarding trees between routers, connecting all subnets containing multicast group members, as illustrated in Figure 2.2. A separate forwarding tree is created for each multicast group. Within subnets, the Internet Group Management Protocol

\(^1\)In practice there are a range of real-time applications ranging from those with loose qualitative QoS requirements to those with strict quantitative requirements. However in all cases these applications need a higher level of service than best effort.
(IGMP) [Dee89], enables hosts to inform their multicast router which multicast groups they wish to join. [SM97] provides a good overview of Internet multicast delivery.

Several multicast routing protocols have been developed to create forwarding or delivery trees. These include: Protocol Independent Multicast Sparse Mode (PIM-SM) [DEF+96, EFH+98], PIM Dense Mode (PIM-DM) [DEF+98], Distance Vector Multicast Routing Protocol (DVMRP) [DPW88, Pus98], Multicast Extensions to Open Shortest Path First (MOSPF) [Moy94a, Moy94b], and Core Based Trees (CBT) [Bal97b, Bal97a, BFC93].

DVMRP, MOSPF and PIM-DM all create Sender based (or Source) Forwarding Trees, representing the shortest path between a sender and receivers (see Figure 2.3(a)). These multicast routing protocols are designed for environments densely populated by group members and where network link capacity is plentiful. The main shortcoming of this multicast approach is that it does not scale to large numbers of senders, or to sparsely populated multicast groups [SM97].

PIM-SM and CBT both create shared multicast forwarding trees (see Figure 2.3(b)). These protocols are designed for WAN environments where link capacity is scarce (or expensive) and where group members are widely distributed. The Shared Forwarding Tree, also termed the sparse mode approach, uses resources more efficiently than Source Forwarding Trees. Indeed, the objective of these proposals is to provide scalable multicast routing over the Internet, [SM97]. The primary disadvantage of this approach is that traffic concentrates at the core of the Shared Forwarding Tree (typically the creator of the multicast group selects the core). Moreover, multicast traffic may follow sub-optimal routes since it must travel via the core node, also known as the Rendezvous Point (RP). Simulation studies have shown that end-end delay is approximately 10% greater when Shared
Forwarding Trees are used, rather than Source Forwarding Trees [SM97]. Although PIM-SM creates Shared Forwarding Trees by default, Source Forwarding Trees can also be created. A receiver can switch to a Source Forwarding Tree when it does not receive sufficient QoS (e.g. end-end delay). In the remainder of the thesis Internet multicast delivery is assumed to use the PIM-SM multicast routing protocol.

2.3.2 Best Effort and Real Time Service Support

Traditionally the Internet provided best effort service only. This is achieved by forwarding datagrams at routers on a first come, first served (FCFS) basis, using whatever resources are currently available.

To support real time applications and allow link capacity to be shared between multiple traffic classes, the Integrated Services model was proposed [She95, SCZ94, Whi97, Wro97b]. Service models, traffic control, packet scheduling and a resource setup protocol are the key components of the Integrated Services model [BCS94]. Traffic control (e.g. [SCZ93]) and packet scheduling (e.g. [JDSZ97]) are vendor dependent mechanisms and hence are not described here. The IETF have developed two new service models [Wro97a, SPG97], in addition to best effort service. Furthermore the ResourceReScrVationProtocol(RSVP) [ZDE+93, MBB+97, BZB+97], has been created to allow unicast and multicast applications to specify their service requirements to the network. RSVP does not determine the path traffic should follow through the network, or what resources each router should
2. The Future Provision of Internet Services

Figure 2.3: Comparison of Source and Shared Multicast Forwarding Trees

(a) Source Forwarding Tree

(b) Shared Forwarding Tree
RSVP is designed to operate in conjunction with admission control, packet scheduling and routing protocols to provide real time Internet services. RSVP is an IETF proposed standard and a number of organisations including ISI, Bay Networks, Cisco, IBM, Intel, Sun and SGI have, or are developing RSVP based products [Uni].

To reserve resources across the network, RSVP uses the One Pass With Advertising (OPWA) mechanism. Senders "advertise" application traffic characteristics in path messages. Routers between senders and receivers modify path messages to describe the service they provide (e.g. how much delay they contribute to the overall end-end delay). Receivers determine their QoS requirements on the basis of the path message contents. The receiver QoS requirements are transmitted to the sender in resv messages. The information contained in resv messages is then used by routers and senders to reserve the requested QoS. This process is illustrated in Figure 2.4.

RSVP is designed to allow resources to be reserved for applications, potentially involving multiple senders and receivers [ZDE+93]. Three design features of RSVP are crucial to support multipoint-to-multipoint (mpt-mpt) communication in a scalable fashion. These features are heterogeneous receiver support, resource sharing and dynamic reservation parameter negotiation.

Heterogeneous receiver support is necessary for two reasons: (1) receivers may have different processing capabilities, and (2) the paths to receivers may have different capacities. RSVP supports heterogeneous receivers, by allowing each receiver to specify different QoS requirements. Figure 2.5 shows how the Internet protocol suite supports two receivers with different maximum end-end delay requirements, assuming receiver 1 will accept a higher end-end delay than receiver 2.
This allows the packet scheduler to interleave packets from different senders when transmitting data to receiver 1, whereas to meet the delay requirements of receiver 2, no interleaving occurs. To avoid the problem where the sender must process a resv message for each receiver, resv messages are merged at routers on the path to the sender as shown in Figure 2.4. At any merge point the most stringent QoS requirements received, are forwarded upstream to the sender. For example, if two receivers request a maximum end-end delay of 100 ms and 500 ms respectively, the 100 ms delay requirement is forwarded to the sender.

In many mpt-mpt applications, only a subset of the potential senders communicate at once. For example, in a voice conference, typically only one user speaks at any time. This means that resources only need to be reserved for the number of simultaneously transmitting senders. This concept is termed resource sharing. RSVP specifies a number of reservation styles to allow receivers to indicate whether the resources reserved should be used by all senders (shared), or whether separate resources must be reserved for each sender (distinct).

Finally, in many applications, senders and receivers periodically join and leave the real time session, for example, in long-lived conferences. Furthermore, during a real time session, the route between senders and receivers may change. To allow for both of these circumstances, path and resv messages flow throughout the lifetime of the application, constantly refreshing the reservation. This allows the reservation parameters for a particular application to be dynamically modified.

As mentioned earlier, the IETF has standardised two real time service classes. Guaranteed service [SPG97], guarantees a maximum end-end delay, and is in-
tended for audio and video applications with strict delay requirements. Controlled Load service [Wro97a], guarantees to provide a level of service equivalent to best effort service in a lightly loaded network, regardless of network load. This service class is designed for adaptive real-time applications (e.g. applications that can modify their play-out buffer as the end-end delay varies). Although only two Internet service classes are currently defined, the Internet protocol suite is sufficiently flexible to allow new service classes to be specified in the future.

An alternative approach for supporting real-time applications within the Internet is being developed within the IETF. This is termed Differentiated Services [BBC+98, BBB+98, NBBB98]. It is believed that Internet Service Providers will not want to provide fine grained resource allocation, and employ signalling protocols such as RSVP. Instead users needs can be satisfied by offering a range of service types from which they can select. This approach can not provide strict real-time guarantees. Instead it provides different levels of best effort service that vary in terms of their dropping and delay priorities. A differentiated services field in the IP datagram header is used to indicate the datagrams service requirements. Traffic conditioning and policy enforcement at network boundaries is essential to implement a range of services. This approach is more in keeping with the traditional Internet connection-less paradigm, however it can only provide loose service guarantees. In contrast the Integrated Services approach can provide strict guarantees to real-time applications. Moreover work is being undertaken to improve the scalability of the Integrated Services approach by aggregating flow information stored at routers [BV98, GA98]. Given that the Integrated Services approach is more mature than the Differentiated Services approach we focus on the Integrated Services approach in the remainder of the thesis.

This section has reviewed how the Internet protocol suite provides (1) unicast and multicast delivery, and (2) best effort and real time service. The next section describes how the ATM protocol suite supports these delivery and service classes.

2.4 ATM Protocol Suite

ATM is a connection oriented protocol suite, designed to support a wide range of applications in a variety of network environments. In particular, ATM can operate over a range of physical media, distances and link rates. ATM also enables the transfer of data at high speeds through the use of small fixed size 53 byte cells. For a good description of ATM refer to [dP91, MS95, DL95].

ATM protocols can be divided into two groups, on the basis of whether they are used over the User Network Interface (UNI), or over the Network Node Interface
2.4.1 Unicast and Multicast Delivery Support

To enable unicast delivery in ATM networks, a point-to-point (pt-pt) VC must be created between the sender and receiver. Figure 2.6 shows the creation of a pt-pt VC via the UNI signalling protocol, but makes no assumption regarding the NNI signalling protocols, which can be vendor specific. Several versions of the UNI have been developed. UNI 3.1 is in widespread use, however UNI 4.0, which was completed in 1996, is becoming increasingly available. As a result the remainder of this thesis, considers both UNI 3.1 and UNI 4.0.

Once the pt-pt VC exists, traffic can flow between the sender and receiver. Figure 2.7 shows how application PDUs are encapsulated in lower layer PDUs for transmission across the ATM network. The ATM Adaptation Layer (AAL) protocol provides an interface between higher layer protocols and the ATM transmission protocol.

AAL 3/4 and AAL 5 are both designed to carry connection-less traffic such as
Figure 2.7: ATM Unicast Data Delivery

Internet traffic over ATM networks. However, AAL 3/4 has much higher per cell overhead and is more complex than AAL 5. This is primarily due to the addition of a multiplexing function that allows cells from different senders to be interleaved. As a result, due to its simplicity and low overhead, AAL 5 is currently the most popular AAL, particularly for Internet applications. As can be seen in Figure 2.7, AAL 5 PDUs are segmented into fixed size ATM cells which are transmitted to the receiver via the pt-pt VC. At the receiver, the AAL 5 PDU is reassembled, the trailer stripped, and the payload passed to the higher protocol layers.

Using current ATM standards, multicast delivery can be provided in several ways. One option is to create a mesh of pt-pt VCs connecting each sender - receiver pair, as shown in Figure 2.8(a). This approach is not widely used because (1) it requires a large number of VCs and (2) senders must transmit multiple copies of each cell, one for each receiver. An alternative is to create a point-to-multipoint (pt-mpt) VC from each sender to all receivers, as depicted in Figure 2.8(b). This approach has the advantage that only one VC has to be maintained per sender. Furthermore, when using pt-mpt VCs, cells are only duplicated when the path to two or more receivers diverges.

To join new receivers to a pt-mpt VC using UNI 3.1, senders transmit an add party signalling message to the ATM network as shown in Figure 2.9(a). The sender then receives an add party acknowledgment message when the receiver
Figure 2.8: Point-to-Point and Point-to-Multipoint ATM Multicast Delivery Options
Figure 2.9: Alternative Approaches for Adding a Receiver to a Pt-Mpt VC
is successfully joined to the pt-mpt VC. This process is termed Root Initiated Join (RIJ). In UNI 4.0, pt-mpt signalling protocols are extended to include Root Leaf Initiated Join (RLIJ), and Network Leaf Initiated Join (NLIJ). Both RLIJ and NLIJ allow the receiver to initiate the request to join the pt-mpt VC (see Figure 2.9(b) and (c)). However, in RLIJ, the sender must still process the receiver join request. In contrast, NLIJ allows the network to join the receiver to the pt-mpt VC without notifying the sender, thus increasing the scalability of pt-mpt VCs.

Another way to provide multicast delivery in ATM networks, is to create a multipoint-to-multipoint VC between the senders and receivers. Current ATM standards do not support mpt-mpt VCs, mainly due to the problem that AAL 5 does not allow cells from different PDUs to be interleaved on the same VC. Researchers are currently considering how to best provide multipoint-to-multipoint VCs [GR96, Tur96, GLO96]. Given the provision of mpt-mpt VCs is a research area in its own right, and that mpt-mpt VCs will not be available for some time, they are not considered further in this thesis.

2.4.2 Best Effort and Real Time Service Support

When a pt-pt or pt-mpt VC is created, the application traffic characteristics and QoS requirements must be specified. The ATM protocol suite provides several service categories which allow an application to specify its traffic characteristics. ATM standards also define several QoS classes which are used to indicate the required QoS. In this section, the service categories and QoS classes provided by ATM standards are described.

In UNI 4.0 networks, the following service categories are provided [The96a, Gar96]:

**Constant Bit Rate (CBR)** Provides delay bounds and no loss by allocating a fixed bit rate.

**real-time Variable Bit Rate (rt-VBR)** Provides delay and loss bounds for bursty applications.

**non-real-time VBR (nrt-VBR)** Provides loss bounds for bursty applications.

**Available Bit Rate (ABR)** Attempts to minimise loss and provide a fair share of the link capacity to applications via flow control.

**Unspecified Bit Rate (UBR)** Provides no service guarantees.
CBR, rt-VBR and UBR are also supported in UNI 3.1 networks [The94]. Within each of the service categories described above, a number of traffic parameters must be specified (e.g. peak cell rate, sustained cell rate, and maximum burst size).

Both ABR and UBR have been designed to support best effort services. However, in core networks, where best effort traffic is highly aggregated, CBR is also applicable. Hence, the volume of best effort traffic being supported must be considered when selecting a best effort service category.

The service category that should be used for real-time applications depends on the exact characteristics of each application. For example, for video playback applications with strict delay requirements the CBR or rt-VBR service categories are most appropriate.

In both UNI 3.1 and 4.0, QoS requirements are indicated by selecting a QoS Class. QoS classes allow a host to specify the broad level of service they require, without needing to specify exact parameters. UNI 4.0 also allows individual QoS parameters (such as the acceptable cell delay variation and cell loss ratio) to be specified. The following QoS classes are currently defined:

QoS Class 0 Unspecified QoS, i.e. provides best effort service.

QoS Class 1 Provides performance comparable to digital private lines.

QoS Class 2 Meets the requirements of packetised audio and video.

QoS Class 3 Used to interoperate with other connection oriented protocols e.g. Frame Relay.

QoS Class 4 Used in conjunction with connection-less protocols e.g. IP.

To summarise, to support best effort or real time applications, the ATM service category and QoS class should be selected that matches the application’s requirements most closely.

2.5 Proposals for Providing Internet Services over ATM

From Sections 2.3 and 2.4 it is clear that both the Internet and ATM protocol suites provide mechanisms to support (1) unicast and multicast delivery and (2) best effort and real time service. These sections have also highlighted the key difference between the two protocol suites. That is, ATM follows a connection-oriented paradigm, whereas the Internet follows a connection-less paradigm. This
The difference in paradigms means Internet services can be provided over ATM in many different ways [CSV96].

In recent years many researcher have been considering how to best provide Internet services over ATM. As a result, several proposals for providing Internet services over ATM have been developed. These proposals are described in more detail in this section.

In 1993, the ITU-T standardised two proposals for supporting Internet services (and other connection-less protocols) over ATM [Tel93a, LMOT94, HS95]. The first proposal, termed Direct Service [Tel93b, ANV96], places Connection-less Servers (CLS) within the network. The connection-less traffic is forwarded across the network from CLS to CLS. The Direct service has become unpopular because it relies on AAL 3/4, while most current networks employ AAL 5 due to its lower complexity and overhead. The second proposal, termed Indirect Service [Tel93a] places the task of connection-less service support on higher protocol layers, with the ATM network providing a transport mechanism only. With this proposal, the onus is placed upon other technical bodies such as the IETF and the ATM Forum to develop the protocols necessary to carry Internet services over ATM. Proposals based on Indirect Service are outlined in the remainder of this section.

The IETF has developed several protocols, which when used together, provide a complete solution to the overall problem of providing Internet services over ATM. A suite of protocols termed Classical IP over ATM (CLIP) have been developed to support unicast best effort Internet applications [LH98, Hei93, PLM+95]. This protocol suite is commercially available and in widespread use. This protocol suite has also been extended to support UNI 4.0 ATM networks [Mah98], and IPv6 [ASJ99]. An alternative form of unicast delivery, termed the Next Hop Resolution Protocol (NHRP) [LKP+98, Can98, RH98], that reduces the number of routers traffic must traverse between the sender and receiver has also been developed within the IETF. A mechanism for transitioning between CLIP and NHRP has also been produced [Luc98].

In 1996, the IETF approved the MARS proposal to support multicast and broadcast best effort applications [Arm96, Arm97a, SA97]. Currently, the IETF Integrated Services over Specific Lower Layers working group (ISSLL), is considering how to best support both unicast and multicast real time Internet services over ATM [Ber98b, Ber98a, GB98, CBB+98]. The solutions developed within the ISSLL working group are described further in Section 2.6.3. It is envisaged that all of the proposals being developed within the IETF will be used together to provide a flexible solution to the problem of providing Internet services over ATM.

The ATM Forum have developed two proposals for providing Internet services
over ATM. The first, termed LAN Emulation (LANE) supports both unicast and multicast best effort services over ATM [The95]. LANEv2 was completed in July 1997 and extends LANEv1 by providing real time services support and more flexible multicast delivery. The ATM Forum have also developed the Multi Protocol Over ATM (MPOA) approach [The97, FRT96]. This combines the NHRP and LANE proposals to develop a more flexible solution for delivering traffic across multiple ATM subnets.

The ATM Forum and IETF have also developed an extension of the ATM Private Network to Network Interface Routing Protocol (PNNI) [The96b]. This allows an IP routing topology to be dynamically overlaid on an ATM network. This technique is termed Proxy PNNI Augmented Routing [For98, DP99].

Over the last two years several manufacturers have developed a new network device that combines an IP router and ATM switch. It is envisaged that the deployment of such a device will allow Internet services over ATM to be deployed in a more flexible manner. A standardisation of this research effort termed Multi Protocol Label Switching (MPLS) is occurring within the IETF [CDF+97, KKV97]. The objective of MPLS is to simplify forwarding by using a short fixed length label to identify a traffic stream rather than processing several header fields. In the ATM context this means forwarding at the ATM layer (on the basis of the VPI/VCI) rather than at the IP layer.

Solutions to the problem of providing Internet services over ATM must support (1) unicast and multicast delivery and (2) real time and best effort service. Table 2.1 categorises proposals in terms of the delivery and service classes they support.

<table>
<thead>
<tr>
<th>Proposal</th>
<th>Delivery Class</th>
<th>Service Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Unicast</td>
<td>Multicast</td>
</tr>
<tr>
<td>Direct Service</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>CLIP</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MARS</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>NHRP</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>ISSLL</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LANEv1</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LANEv2</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>MPOA</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Proxy PAR</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>MPLS</td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 2.1: Classification of Proposals in Terms of Delivery and Service Class.

Many of the proposals presented in this section employ the same underlying approach to provide Internet services over ATM. Hence this thesis focuses on
the alternative approaches for providing Internet services over ATM, rather than upon the details of specific proposals. The next section describes the key issues that must be resolved to provide Internet services over ATM. The remainder of the chapter focuses on alternative approaches, indicating which proposals employ each approach.

2.6 Key Issues

The problem of providing Internet services over ATM, can essentially be decomposed into the following three key issues:

- Determining the ATM address of each receiver, or the address of an edge router if the receiver is not attached to the ATM network,
- Deciding what VCs to create to allow sender(s) to communicate with receiver(s), and
- Ensuring that sender(s) and receiver(s) obtain the required level of service.

The remainder of this section, describes each issue, and discusses how it has been addressed in the current literature.

2.6.1 Address Resolution

To communicate over an ATM network, the IP address of a receiver must be mapped to an ATM address. Traditionally, the Address Resolution Protocol (ARP), performs this function by broadcasting a request, containing the receiver’s IP address, over the local subnet [Plu82]. On receipt of the request, the owner of the IP address responds with its data link layer address. However, ATM does not naturally provide a broadcast capability. This means that either a broadcast topology must be emulated or alternative address resolution mechanisms need to be developed. Figure 2.10 compares two emulated broadcast approaches (one employing pt-pt VCs and the other pt-mpt VCs) with an address resolution server (ARS) approach.

Emulated broadcast allows ARP to be used unchanged. However, this approach means senders must transmit an ARP request to every subnet host. This requires both a large number of VCs and ARP request duplication, especially when pt-pt VCs are employed. Furthermore, UNI 3.1 networks do not support multicast group addresses. Hence, resolving an IP multicast address using ARP, requires all group members to respond to the request. This creates the additional problem of deciding when the sender has received all of the responses.
(a) Emulated Broadcast Approaches

(b) Address Resolution Server

Figure 2.10: Alternative Address Resolution Mechanisms
Due to the poor performance of ARP, most current proposals (e.g. CLIP, LANE, NHRP, MARS and MPOA), employ an Address Resolution Server (ARS). The ARS approach has the advantage that only one address resolution request and response needs to be transmitted. This is regardless of the number of subnet hosts or whether a multicast address is being resolved. Furthermore, each host only needs to submit its address information occasionally, rather than responding to a large number of requests. To overcome the problem where the ARS overloads, the IETF have developed the Server Cache Synchronisation Protocol (SCSP) [LAHD98], to support multiple ARSs. Hence the current literature favours the use of ARSs to provide address resolution in ATM networks as indicated by its use within LANE, NHRP, MARS and MPOA.

2.6.2 Delivery Mechanisms

To transmit data to a receiver or edge router on the path to the receiver, one or more VCs must be created. There has been much debate on how the communication path between senders and receivers should be created. This has focused on whether the communication path, or delivery mechanism should follow: (1) the Internet connection-less model, (2) the ATM connection oriented model, or (3) a hybrid approach combining the connection-less and connection oriented models.

Many proposals have appeared in the literature for providing unicast and multicast delivery of Internet applications over ATM. However, there is still no clear consensus of which approach should be followed. Clearly, this issue is central to the whole problem of providing Internet services over ATM. This is because many other issues including the scope of address resolution protocols and real-time service support depend on the delivery mechanism selected. This thesis addresses this delivery mechanism issue, with the objective of determining which delivery mechanisms are best suited for carrying unicast and multicast Internet applications over ATM.

2.6.3 Providing the Required QoS

When supporting real time applications, the data must not only arrive at the destination, but arrive with the required QoS. There are two major aspects of this problem [CWSA95]. The first issue is the translation of Internet service class parameters to ATM service class parameters. The second issue is the inter-operation between RSVP and ATM signalling protocols. This is necessary to communicate application traffic characteristics and QoS requirements to switches and routers on the path between senders and receivers. The IETF Integrated Services over
Specific Lower Layers (ISSLL) and the ATM Forum are both currently attempting to address these two sub-problems.

Parameter Translation

Significant progress has been made on determining how to translate Internet service class parameters to ATM service class parameters. [PLM+95] and [Mah98] describe how to support best effort applications in UNI 3.1 and UNI 4.0 networks respectively. Garrett and Borden, [GB98], describe how all three currently defined Internet service classes, (best effort, guaranteed (GS) and controlled load (CL)), should be mapped to ATM service categories and QoS classes, in both UNI 3.1 and UNI 4.0 networks.

The service category suitable for a given VC is highly dependent on the characteristics of the traffic that it will carry. For GS, CBR or rt-VBR should be used depending on the burstiness of the traffic. Both ABR and nrt-VBR are well suited for CL. However, ATM standards currently do not support ABR pt-mpt VCs, which means nrt-VBR should be used for CL multicast applications. Best effort service is best provided by CBR in the core of a network, when a high level of aggregation occurs. If traffic volumes are lighter, UBR or ABR are recommended. Significant progress has been made in resolving the issue of parameter translation, and hence this issue is not considered further in this thesis.

Communication of Resource Requirements

To ensure that an application’s resource requirements are met, mechanisms are required to map RSVP to ATM signalling. As discussed in Section 2.3, three important features of RSVP are the ability to: (1) support heterogeneous receivers, (2) share resources between senders, and (3) dynamically modify the reservation parameters. Approaches for mapping RSVP to ATM signalling must support these three features. Mechanisms for providing these capabilities are described in [OS96, Ber96, CBB+98]. Moreover, work is progressing within the ATM Forum and ITU-T organisations to extend the ATM protocol suite to better support these capabilities.

Approaches for providing each of the real time capabilities are briefly outlined below. This discussion focuses on the best currently available approach and how the ATM protocol suite can be extended to better support each feature.
2. The Future Provision of Internet Services

Heterogeneous Receiver Support

As described in Section 2.3, heterogeneous receiver support allows receivers for the same real-time session to request different levels of QoS. In the ATM protocol suite, pt-mpt VCs are the key mechanism for delivering traffic to multiple receivers. Heterogeneous receivers can easily be supported in an ATM environment by allowing different leaves of the pt-mpt VC to have different characteristics. However, these variegated VCs, are not supported by current ATM standards. Both the ATM Forum and ITU-T are considering how to support variegated VCs. However, alternative mechanisms are required to support heterogeneous receivers until variegated VCs are standardised.

[Ber98b] and [Ber98a] recommend that current networks use a hybrid approach that delivers traffic to all receivers with the most stringent QoS by default. However, if one or more receivers can not accept this level of service, a best effort VC should be created to ensure that the receivers can still participate in the real time session. This approach ensures each receiver can join the real-time session, but only requires at most two VCs per session.

Resource Sharing

Resource sharing allows multiple senders to use the same set of network resources. Ideally, this capability would be provided in an ATM network by creating a mpt-mpt VC connecting all senders and receivers. This allows resources to be shared between senders as soon as their paths to receivers overlaps. However, as mentioned earlier, ATM standards currently do not support mpt-mpt VCs. This means alternative approaches must be used to support resource sharing in existing ATM networks.

One simplistic approach is to create a separate VC for each sender. However, this allows no resource sharing, and thus wastes network resources, particularly when only a small percentage of senders transmit data simultaneously. In current ATM networks, some degree of resource sharing can be achieved by aggregating traffic either at servers or routers within the network. Using this approach, separate resources are reserved from senders to the aggregation point. However, from the aggregation point to the receivers, resources can be shared. Care must be taken when selecting the location of the aggregation point, to maximise the degree of resource sharing that can be achieved.
Parameter Renegotiation

As discussed in Section 2.3, RSVP also supports the dynamic modification of reservation parameters. This capability is termed parameter renegotiation. Both the ATM Forum and ITU-T, are currently developing standards to support in-call renegotiation. This capability will allow VC parameters to be dynamically modified. However, until in-call renegotiation is standardised, alternative approaches to parameter renegotiation must be used. [CBB+98, Ber98b] and [Ber98a] recommend that parameter renegotiation should be emulated in current ATM networks by (a) creating a VC with the new QoS requirements, then (b) switching the traffic to this VC, and (c) tearing down the old VC. To limit the load on the signalling network [CBB+98] also recommends that the frequency of parameter renegotiation requests for one real-time session be limited by a timer.

Summary

It is clear that good progress is being made in the problem of mapping RSVP to ATM signalling. As the ATM protocol suite is extended to support variegated VCs, mpt-mpt VCs and in-call renegotiation, ATM will easily support all of the features of RSVP. In the meantime, [CBB+98, Ber98b] and [Ber98a] have recommended approaches that should be used to support heterogeneous receivers, resource sharing and parameter renegotiation in existing ATM networks. As a result, the mapping of RSVP to ATM signalling is not considered further in this thesis.

Given the Differentiated Services approach is still in the early stages of specification, little work has been performed to consider how to map Differentiated Services to ATM networks. In broad terms, a separate VC could be created for all of the differentiated service classes supported by a given network. The characteristics of this VC will depend on the header field values. Given much work needs to be performed before the Differentiated Services approach is specified, it is not considered further within this thesis. However, it appears that this approach can be mapped to an underlying ATM network in a straightforward manner.

2.7 Conclusions

In this chapter it has been shown that although both the Internet and ATM protocols suites are designed to support applications with a variety of delivery and service requirements, they do so very differently. As a result, Internet services can be provided over ATM networks in many different ways, depending on whether the
connection-less, connection oriented, or a combination of both of these paradigms is followed.

The discussion in this chapter has also described the three key problems that must be resolved when providing Internet Services over ATM: (a) address resolution, (b) delivering the traffic, and (c) ensuring QoS requirements are met. This has highlighted the fact that the selection of a delivery mechanism is central to the overall problem of providing Internet services over ATM. Furthermore, there is no clear consensus in the existing literature of which delivery approach should be followed. As a result the remainder of this thesis focuses on the performance comparison of approaches for providing both unicast and multicast Internet traffic delivery over ATM.
Chapter 3

Providing Internet Services over ATM

3.1 Introduction

As discussed in Chapter 2, the selection of a delivery mechanism is central to the problem of providing Internet services over ATM networks. Hence the remainder of the thesis focuses on approaches for delivering Internet traffic over ATM. A review of unicast delivery approaches, and literature investigating the performance of these approaches, is found in Section 3.2. This is followed by a similar review for intra-subnet and inter-subnet multicast delivery approaches in Sections 3.3 and 3.4 respectively. Section 3.5 then describes alternative multicast delivery systems in light of the discussion in Sections 3.3 and 3.4. Section 3.6 presents the shortcomings apparent in the current literature describing how to provide Internet services over ATM. This chapter then concludes with a description of how this thesis addresses these shortcomings.

3.2 Unicast Delivery Approaches and their Analysis

Currently, most Internet applications involve the interaction of a single sender and receiver, and hence require unicast delivery (e.g. WWW, FTP and Telnet). As a result it is crucial that any solution for providing Internet services over ATM supports unicast delivery efficiently. This section describes alternative approaches for supporting unicast Internet applications over ATM, highlighting the shortcomings of literature in this area.

As discussed in Chapter 2, the Internet protocol suite traditionally employs a connection-less delivery mechanism where data is transmitted from router to
router towards the destination. In contrast, the ATM protocol suite follows a connection oriented paradigm where VCs must be created before data transmission commences. Approaches for providing unicast Internet delivery over ATM tend to follow one, or a combination of these two delivery paradigms.

3.2.1 Hop-by-Hop Approach

One popular delivery mechanism is the Hop-by-Hop approach. Both CLIP and LANE follow this approach which is illustrated in Figure 3.1(a). The Hop-by-Hop approach retains the traditional Internet structure where the network is comprised of subnets interconnected by routers. When the sender and receiver reside on the same subnet, unicast delivery is provided by creating a pt-pt VC between them. However, if the sender and receiver reside on different subnets, the sender must forward data to its subnet router via a pt-pt VC. The subnet router then reassembles the ATM cells, and routes the resulting IP datagram towards the

---

**Figure 3.1: Alternative Inter-Subnet Unicast Topologies**

- (a) Hop-by-Hop
- (b) Cut-Through
- (c) Label Switching
3. Providing Internet Services over ATM

destination subnet. When the destination subnet router receives the unicast traffic it forwards the data to the receiver via a pt-pt VC (see Figure 3.1(a)).

Another variation of the Hop-by-Hop approach is to employ pt-pt VCs between routers connected via an ATM backbone network. For example to create a full mesh of VCs between all routers connected to that ATM backbone. Proxy PAR [For98, DP99] is an example of an IP over ATM technique that falls within this category of hop-by-hop approaches.

3.2.2 Cut-Through Approach

Many researchers argue that the Hop-by-Hop approach does not fully utilise the capabilities of the ATM protocol suite [BPR94, RK96]. As discussed in Section 2.4, like the Internet, ATM is also designed to be a global communications protocol. This means that hosts in different ATM administrative domains can communicate directly using ATM level protocols if a pure ATM communication path exists between them. The Cut-Through approach achieves this by creating end to end direct VCs between hosts when an ATM path exists between them (see Figure 3.1(b)). The Cut-Through approach has the additional benefit that better routing decisions can be made because the end to end path is now determined solely using ATM routing protocols, rather than a combination of IP and ATM routing protocols [BPR94].

Three proposals have appeared in recent years employing the Cut-Through approach: NARP [HG94], NHRP [Can98, LKP+98] and MPOA [The97]. If both sender and receiver are attached to the ATM cloud, creating a direct VC is straightforward. All the sender needs to determine is the ATM address of the destination. NARP, NHRP and MPOA describe mechanisms to achieve this. Furthermore, when either, or both, the sender and receiver are not attached to the ATM cloud, it is still desirable to minimise the number of routers that traffic passes through. Both NHRP and MPOA support this by creating a direct VC between the ingress and egress routers closest to the senders and receivers [LKP+98]. However, when direct VCs are created across transit ATM networks (i.e. when neither the sender or receiver reside on the ATM network) persistent routing loops can occur [CSV96]. As a result, the current NHRP and MPOA proposals are limited to cases where the receiver is directly attached to either the ATM cloud, or to an ATM egress router. Extensions to the NHRP proposal are being developed to overcome this limitation [RH98].
3.2.3 Comparison of Hop-by-Hop and Cut-Through Approaches

The Cut-Through approach produces lower end-end packet delays than the Hop-by-Hop approach because reassembly no longer occurs within the ATM network. However, the VC and signalling resource requirements of the Cut-Through approach are greater than the Hop-by-Hop approach. This is because each sender-receiver pair requires a separate VC (see Figure 3.1(b)). In contrast, the Hop-by-Hop approach aggregates all traffic destined for the same next hop router on one outgoing VC (see Figure 3.1(a)). The lower Hop-by-Hop VC requirements also translate to lower signalling resource requirements. This occurs because aggregate VCs between routers are more likely to exist than a VC between a specific sender and receiver. Furthermore, in failure situations e.g. where an ATM link fails, the Cut-Through approach will require more resources to recover. This is because there will be a greater number of VCs to re-establish (one per sender-receiver pair using the failed link), compared to the Hop-by-Hop approach where all traffic carried on a given link can be aggregated onto a single VC. To summarise, the Cut-Through approach is not only more complex than the Hop-by-Hop approach, but it has higher VC and signalling resource requirements.

It would appear that the Cut-Through approach will always produce lower response times than the Hop-by-Hop approach. However, if the direct VC must be created this may no longer be the case. That is, it may be faster to send the data hop-by-hop than to first create the direct VC and then transmit the data. Furthermore, if the direct VC must be created, the sender may not know the ATM address of the receiver or egress router. Hence, address resolution may be required before the direct VC can be created. As illustrated in Figure 3.2, address resolution can require requests to be transmitted across the entire ATM network.
network. Hence there can be a significant delay penalty associated with resolving the address, compared to the Hop-by-Hop approach where address resolution only occurs within subnets. Addresses can however be cached at intermediate Next Hop Servers (NHSs) or hosts to reduce the Cut-Through approach address resolution delay.

Once the ATM address of the receiver or egress router is known, the direct VC can be created. Typically the decision to create a direct VC will be triggered by the generation of data at a sender, (or arrival of data at an ingress router). If the direct VC does not exist, a mechanism is needed that manages the data that arrives until the direct VC is opened. Three possible approaches have appeared in the literature [LKP+98]. The first is to drop any traffic that arrives before the VC exists, and thus requires this traffic to be retransmitted. Another possibility, termed the Buffered approach, is to buffer data at the sender or ingress router until the VC opens. The third option, termed the Hybrid approach, forwards the data using the Hop-by-Hop approach until the direct VC is created. At that time the data transmission can be switched over to the new VC. This is the currently recommended approach [LKP+98].

It seems likely that the Hybrid approach will provide the best performance because data can be forwarded even though the direct VC does not exist. However, no performance analysis has appeared in the literature, showing this to be the case. If the Hybrid approach is followed and it takes a long time to create the VC, or the volume of data being transmitted is small, all of the data may have been transmitted hop-by-hop, before the direct VC is ready. In this case, signalling resources are wasted by creating a redundant direct VC. Clearly, performance analysis is required to compare the performance of the Hop-by-Hop and Cut-Through approaches, particularly when the Cut-Through approach must create the direct VC.

3.2.4 Label Switching Approach

One of the main disadvantages cited with the Cut-Through approach is the delay associated with creating the direct VC. To overcome this problem, several organisations including Ipsilon [NML98, NMLH97, NLM96, Ips96, NEH+96a, NEH+96b], Cisco [RDR+97, KKV97], Toshiba [KNME97], NEC [ADA97, ALAR98], IBM [BBD+97] and Washington State University [PST95a, PST95b], have developed proposals that use a combination of the Hop-by-Hop and Cut-Through approaches, termed Label Switching (see Figure 3.1(c)). The IETF have also formed the Multi-Protocol Label Switching working group to consider this approach [CDF+97, RVC98]. Like the Hop-by-Hop approach, Label Switching retains routers in the
3. Providing Internet Services over ATM

interior of the ATM network. However Label Switching proposes a new type of router, termed the Label Switching Router (LSR) which combines traditional IP routing functions with an ATM switch. In some LSR schemes (e.g. IP Switching) each LSR traffic can be: (1) reassembled, routed and forwarded to the next hop like the Hop-by-Hop approach (routed), or (2) forwarded directly through the switch like the Cut-Through approach (switched). In other schemes (e.g. MPLS), the LSR runs an IP routing protocol, and forms part of the IP routing (e.g. BGP or OSPF) topology, however the traffic is actually forwarded at the ATM layer.

In LSR techniques such as IP switching, flow classification schemes are employed to determine whether a flow should be routed or switched. Furthermore, in IP Switching each LSR can decide independently whether a direct VC should be created for a given flow. If all of the LSRs decide to create a direct VC, then no reassembly will occur in the network. In other techniques such as MPLS, routing topology information is used to create the direct VCs, and all ATM based LSRs on the routing path will forward at the ATM layer, again meaning no packet reassembly will occur in the network. Hence the time to transmit data across the network will often be the same for both LSR and Cut-Through approaches. Therefore, these approaches will potentially only differ in the time to create the direct VC.

3.2.5 Comparison of Cut-Through, Label Switching and Hop-by-Hop Approaches

The Cut-Through approach creates end-to-end direct VCs using standard ATM signalling. In contrast, the Label Switching approach creates the direct VC in a hop by hop fashion as illustrated in Figure 3.3. Although the Label Switching approach is motivated by the argument that the Cut-Through approach creates direct VCs too slowly, no quantitative analysis has appeared in the literature to show: (1) the Cut-Through approach is too slow, and (2) that the Label Switching approach significantly reduces the time to create the end-end cell switched path. Clearly, such analysis is required to determine whether network operators should use the Cut-Through or Label Switching unicast delivery approaches.

Many organisations have proposed Label Switching based approaches. The primary difference between these approaches is (a) how they aggregate the traffic onto the direct VCs, and (b) how they create the direct VCs. At one extreme Label Switching must create a separate VC for each sender-receiver pair, like the Cut-Through approach, for instance if the Label Switching protocol extends to the hosts (see Figure 3.1). This is because cells from different AAL 5 PDUs can not be interleaved on the same VC. Some researchers argue that the Cut-Through and this variant of the Label Switching approach will not scale suffi-
Figure 3.3: Creation of an End-End Switched Path using Label Switching
ciently [Arm97b, KNME97]. As a result, variants of the Label Switching approach have been proposed that enable traffic from different senders to be aggregated onto a single VC.

Aggregation can be achieved by creating one VP for each destination and assigning each sender a separate VC within the VP. However, the VP UNI address space is limited to 8 bits, and many network providers want to use VPs for other functions. Hence this approach has not been popular.

Aggregation is also possible by forwarding all cells comprising an AAL 5 PDU as one burst [CDF+97]. This allows traffic from different senders to be carried on the same VC. The cells comprising an AAL 5 PDU can be grouped using AAL 5 Segmentation and Reassembly (SAR), or by buffering cells until the entire AAL 5 PDU arrives. This means each LSR only needs to maintain one outgoing VC per receiver, rather than one outgoing VC per sender-receiver pair. However, since cells can not be forwarded until the whole AAL 5 PDU arrives, the response times of this VC merging approach will be higher than the Cut-Through and non-VC merging Label Switching approaches. Indeed, if all LSRs perform VC merging, the response time performance will be similar to the Hop-by-Hop approach. This is because the only performance advantage of the VC merging Label Switching approach in this case is that routing is not necessary at each LSR. However given many modern routers cache routing decisions, even this difference begins to disappear. There will also be no difference between the Hop-by-Hop and Label Switching response time performance if each LSR decides to route, rather than switch a given flow.

Label Switching approaches also differ in how they create the direct VC. Some approaches (e.g. Ipsilon and Toshiba proposals) use the arrival of traffic to trigger direct VC creation, and employ a lightweight signalling protocol to achieve this. Other approaches (e.g. the NEC proposal) still use the arrival of traffic to decide a direct VC is needed, but no signalling protocol is employed. Other approaches (e.g. the Cisco proposal) create direct VCs on the basis of topology information, and thus the direct VC may already exist before any traffic is forwarded. Hence there is great variation in how Label Switching approaches create the direct VC.

Researchers claim one of the key advantages of the Label Switching approach compared to the Cut-Through and Hop-by-Hop approaches is that Label Switching is more flexible because (1) some flows can be routed and other switched, and (2) LSRs can independently decide whether to route or switch a flow. However, there is no reason why proposals for unicast delivery can not offer both Cut-Through and Hop-by-Hop unicast delivery. Indeed, the IETF intend the CLIP and NHRP approaches to be used together. Similarly, MPOA, currently being developed by
3. Providing Internet Services over ATM

the ATM Forum provides this dual capability. This means that the first flexibility argument no longer holds. Similarly there is no reason why a traffic flow can not be partially switched and partially routed using the Hop-by-Hop and Cut-Through approaches. Indeed the NHRP specification [LKP+98] states that the direct VC can be terminated part way across an ATM cloud, for example on an administrative boundary where a firewall is required.

This section has described three approaches for Internet unicast delivery over ATM: Hop-by-Hop, Cut-Through and Label Switching. This review has identified the major shortcoming of current literature in this area, which is the lack of analysis comparing the performance of these approaches. In particular, both the Cut-Through and Label Switching approaches were proposed to reduce the delay associated with transporting traffic via the Hop-by-Hop approach. However no analysis has been presented to show that firstly the Hop-by-Hop approach delay performance is too high, and secondly to show that either the Cut-Through or Label Switching approaches significantly reduce this delay. Similarly no analysis has appeared in the literature to show that the Cut-Through approach direct VC creation times are too high, or that the Label Switching approach significantly reduces these delays. Moreover, although it is clear that the Cut-Through and Label Switching approaches have higher VC and signalling requirements than the Hop-by-Hop approach, no analysis has been performed to indicate when the resource usage of these approaches becomes a significant problem. To allow network operators to decide which unicast delivery approach(es) to employ, such analysis is vital. Finally, the Buffered, Retransmission and Hybrid variants of the Cut-Through approach have been proposed for handling data while the direct VC is created. Researchers recommend the Hybrid approach, however, no analysis has been presented to show its performance is superior to the Buffered or Retransmission approach.

3.3 Intra-Subnet Multicast Delivery

In recent years applications requiring the interaction of many senders and receivers have gained popularity (e.g. conferencing, CSCW, DIS and multi-player games). This section reviews delivery mechanisms proposed in the literature to support multicast Internet applications within ATM subnets.

As described in Section 2.3, the Internet provides multicast delivery using multicast routing protocols, IGMP and link layer multicast protocols. In the previous section it was observed that the key difference between unicast delivery approaches is how traffic is transmitted between neighbouring ATM subnets. When consider-
ing multicast delivery this issue must also be addressed. Within subnets, unicast delivery is always provided by creating a pt-pmt VC between the sender and receiver. However, there are several ways to provide multicast delivery within a subnet using a combination of pt-pt and pt-mpt VCs. Hence approaches for inter-subnet and intra-subnet multicast delivery must be considered. Furthermore, the Internet employs two forms of multicast routing protocols, the Source Forwarding Tree and the Shared Forwarding Tree approaches. Therefore the performance of the complete multicast delivery system must be considered. Sections 3.4 and 3.5 continue with an examination of inter-subnet delivery approaches and the multicast delivery system as a whole.

3.3.1 Alternative Intra-Subnet Multicast Delivery Approaches

Most current subnets (e.g. Ethernet and Token Ring) are based on broadcast technology, making it easy to forward multicast traffic from a sender to several receivers. ATM does not naturally provide a broadcast capability and thus alternative multicast delivery approaches must be developed. This section assumes that senders and receivers are attached to the same subnet. However, the discussion equally applies to transit ATM subnets (i.e. when one or more senders or receivers are not directly attached to the ATM network).

To provide multicast delivery within an ATM subnet, the following approaches have been proposed, as illustrated in Figure 3.4:

**Mesh of Pt-Pt VCs (Pt-Pt Mesh)** Creates a pt-pt VC between each sender and receiver (Figure 3.4(a)).

**VC Mesh** Creates a pt-mpt VC from each sender, connecting itself to all receivers (Figure 3.4(b)).

**Multicast Server (MCS)** Creates a pt-pt VC from each sender to the MCS, and a pt-mpt VC from the MCS to all receivers (Figure 3.4(c)).

The VC requirements of these approaches are compared in Table 3.1. This assumes one multicast group containing M senders and N receivers. The approaches are compared on the basis of their: overall VC requirements (Total VCs); the number of VCs each sender originates (VCs per Sender) and each receiver terminates (VCs per Receiver); and the number of new VCs or leaves required to add a new sender (VCs to add Sender) or receiver (VCs to add Receiver).

Table 3.1 shows that in all categories the Pt-Pt Mesh approach has higher VC requirements than all other approaches. This is because a separate VC must
Table 3.1: VC Requirements of Intra-Subnet Multicast Delivery Approaches
be created for each sender-receiver pair. The Pt-Pt Mesh approach also has the
greatest bit rate requirements because senders must transmit N copies of each
packet, one for each receiver. In contrast, the other approaches only duplicate data
when the path to two or more receivers diverges. As a result of the high bit rate
and VC requirements of the Pt-Pt Mesh approach, this approach is not used by
any proposals in the literature. Therefore, the Pt-Pt Mesh intra-subnet multicast
delivery approach is not considered further in this thesis. In contrast, both the
VC Mesh and MCS approaches are employed in the literature. Indeed the MARS
approach [Arm96] supports both the VC Mesh and MCS approaches. Hence the
remainder of this section focuses on the difference between these approaches.

The MARS approach has been designed to support RFC1112 [Dee89] style
multicast (including IGMP). The main objective of IGMP is to track multicast
group membership. Furthermore it is designed assuming multicast is 'cheap'. As
seen above, this is not the case in ATM networks. In IGMP two mechanisms are
of concern: the first is that multicast routers periodically multicast a query to all
multicast hosts, and expect one report back for each multicast group one or more
hosts are members of. This potentially requires all members of multicast groups
to occasionally have to transmit an IGMP report. The other IGMP procedure is
that each receiver must transmit an IGMP report each time they join a multicast
group. [Arm96] proposes an approach where the first query/report mechanism
can be avoided by routers employing the MARS.GROUPLIST_REQUEST and
MARS.GROUPLIST_REPLY messages to obtain the group membership informa­
tion from the MARS rather than querying all multicast hosts. In this thesis we
assume the multicast routers have this capability.

The other issue is how to avoid each host sending an IGMP report each time
it joins a group. The presence of this mechanism requires each host to create
VCs to allow it to send the IGMP message to all other group members in the
local subnet. If the standard IGMP host procedure is maintained the impact will
be: (a) for the VC Mesh approach that all hosts within a subnet will have a pt-
mpt VC to all members of that multicast group (rather than just the senders);
and (b) for the MCS approach that all hosts will have a pt-pt VC to the MCS
(rather than just the senders). These VCs will timeout (the default timer value
is 20 minutes [Arm96]), but the presence of this IGMP procedure will consume
VC and signalling resources. One way to avoid this is for the IP/ATM layer in
each host that implements the MARS client protocol to trap the IGMP messages
and not send them. This is possible because the MARS_JOIN and MARS_LEA

messages provide the equivalent function of informing multicast routers (and any
other senders) of the arrival or departure of a multicast group member). We
3. Providing Internet Services over ATM

assume such an optimisation is possible, and thus do not account for the presence of IGMP report messages in the remainder of this thesis. However this issue of optimising the performance of IGMP over ATM networks does require further consideration.

3.3.2 Comparison of the VC Mesh and MCS Approaches

Table 3.1 highlights the benefits of the MCS approach, particularly in large multicast groups. Firstly, each sender and receiver only manages one VC, regardless of the number of multicast group members. Moreover, the MCS approach creates only one new VC or leaf to add a sender or receiver respectively. This also means that the signalling resources and the time required to add senders or receivers are both lower than for the Pt-Pt Mesh and VC Mesh approaches. Another benefit of the MCS approach is that to join the multicast group, senders require the ATM address of the MCS only, regardless of the number of receivers. In contrast when using the VC Mesh approach in conjunction with UNI 3.1, (which provides no group addressing capability), senders require the ATM address of every receiver.

Although the MCS approach scales well in terms of VC and signalling resource requirements, it has a number of problems. Firstly, the MCS can become a bottleneck because all multicast traffic travels via the MCS. This can be overcome by deploying multiple MCSs [TA96a, TA97a, TA97b, Tal97], however this adds to the cost and the complexity of the MCS approach. Different MCSs can serve: (1) different multicast groups; (2) different senders from the same group; or (3) different receivers from the same group. The first option reduces the load on individual MCSs, without affecting the VC requirements of senders and receivers. Option two also reduces the load on MCSs, however it increases the number of VCs receivers must terminate to one per MCS. The only benefit of the third option is to reduce the number of leaves comprising pt-mpt VCs, however, it requires senders to transmit to multiple MCSs. Thus this approach is only recommended to provide fault tolerance [TA97b].

The primary disadvantage of the MCS approach is that cells must be reassembled into AAL 5 PDUs at the MCS. This is because cells from different AAL 5 PDUs can not be interleaved on the same VC. This delay can not be avoided, regardless of the number of MCSs deployed. Furthermore, if an MCS is serving several multicast groups, IP level processing is required to determine the destination multicast group of each IP datagram. Furthermore, mechanisms are required to allow combined senders/receivers to detect packets which they transmitted, since they will be reflected back from the MCS since the same forwarding VC is used for traffic from all senders. This is termed the 'reflected packet' prob-
The location of MCSs will also affect multicast transfer delay since all traffic must travel via the MCS rather than directly from senders to receivers. Hence the placement of the MCS is crucial.

The VC Mesh approach [Arm96, FMR98, Smi96], does not suffer from the delay performance problems of the MCS approach because each sender creates a separate pt-mpt VC to the receivers. Furthermore, traffic follows the shortest path between senders and receivers, smoothing bit rate requirements across the network. In contrast the MCS approach concentrates traffic at the MCS. Moreover, in the MCS approach traffic follows sub-optimal routes between the senders and receivers because all traffic must travel via an MCS. Hence, although the VC Mesh approach has greater VC and signalling resource requirements than the MCS approach (see Table 3.1), it will provide lower multicast data transfer delays and distribute traffic more evenly across the network. Therefore, there is a trade-off between the MCS and VC Mesh approaches in terms of throughput, congestion, delay, and resource consumption [Fly95, BMM95, Arm96]. It is generally recommended that the MCS approach should be used when VC resources are limited or for dynamic multicast groups, (i.e. where senders and receivers frequently join or leave the group). However, the literature provides no guidelines for values of these parameters where the choice of an approach produces a significant difference in performance.

3.3.3 Performance Analysis of Intra-Subnet Approaches

The key shortcoming in the area of intra-subnet multicast delivery is the lack of literature providing quantitative guidelines of when to apply the VC Mesh and MCS approaches. Indeed, given the MARS scheme offers both VC Mesh and MCS approaches, it is currently left to network operators to determine which approach is appropriate for their particular environment.

Although some analysis has been performed, this has focused primarily on the VC requirements of the approaches [Arm97b, TA97b]. Furthermore, much of the VC requirements analysis has considered only worst case requirements where all hosts are both senders and receivers of all multicast groups [Arm97b, TA97b]. Some analysis of the VC requirements of existing MBone multicast traffic has been performed. However as stated by the authors themselves, current multicast traffic involves only a small number of senders and hence there is little difference between the VC Mesh and MCS approaches [TA97b, Tal97]. Furthermore, the analysis in [TA97b] favoured the MCS approach because only the number of pt-mpt VCs were considered (i.e. the pt-pt VC requirements were ignored). Given the MCS approach requires a pt-pt VC from each sender to the MCS, whereas the
3. Providing Internet Services over ATM

VC Mesh approach requires no pt-pt VCs this biases the analysis.

[TA97b, Tal97] also attempt to analyse the VC requirements of future mixes of multicast applications. However no justification for the assumed application mix was provided. Furthermore, this analysis assumed a maximum of ten senders in a single multicast group. However, applications such as DIS and multi-player games, which are currently gaining much interest will involve much higher numbers of senders than those assumed. This showed hosts need to support around 30% more VCs if the VC Mesh approach is employed compared to the MCS approach [Tal97].

The main difference in the resource requirements of the VC Mesh and MCS approaches is the signalling resources required to add new senders and receivers. This is particularly important at present, because signalling resources tend to be more expensive than VCs in current ATM switches. For instance, current NICs support 1000 VCs [TA97b], but even high end, backbone switches can only create 300 VCs/s (although they can support 100,000 simultaneous VCs) [Tec98]. Clearly, quantitative analysis comparing the signalling resource requirements is essential to enable network operators to select the intra-subnet delivery mechanism best suited to their environment. Such analysis has not been presented in the current literature.

Some delay analysis of the VC Mesh and MCS approaches has appeared in the literature [TAA96, TA97b]. However, this analysis is based on experiments over a simple network containing two switches, a single receiver and six senders. Furthermore, the conclusions drawn from this analysis are misleading in several respects. Firstly, the analysis assumes the VC Mesh approach requires smaller subnets, and hence more mrouters (Internet multicast routers) to connect them than necessary for the MCS approach. This is based on the assumption that the VC requirements of the VC Mesh approach are sufficiently high, that the subnet must be sub-divided. This assumption is not substantiated by the current literature. As a result of the higher number of mrouters, the analysis indicates the VC Mesh approach is likely to produce higher end-end delays than the MCS approach. However, the additional delay is due entirely to the higher number of mrouters. Moreover, the analysis unfairly favours the MCS approach, by only considering small application PDUs and assuming the MCS does not perform any IP level processing, whereas each mrouter must process each IP datagram (thus the MCS and mrouter processing delays were found to be 8.5 ms and 16 ms respectively). In practice MCSs are likely to serve multiple groups and hence must perform IP level processing.

Furthermore, when examining delay [TA97b] the analysis only considers the time for 20 Mbytes of data to arrive at the receiver even though it is comprised of
data sent by many different senders, rather than considering when 20 Mbytes from
a given sender arrives at the receiver, which is the delay users actually perceive.
Hence both the VC Mesh and MCS approach delays will be higher in practice
than the experiments indicate. Indeed, the results show that the delay for the VC
Mesh approach decreases as the number of senders increases, which clearly will
not be the case in practice.

This review has highlighted the need for a detailed quantitative comparison
of the MCS and VC Mesh intra-subnet multicast delivery approaches. In partic­
ular, no analysis has appeared in the literature examining the signalling resource
requirements, or the time required to add new senders or receivers. Furthermore,
the delay analysis that has been published considers only small subnets and small
multicast groups. Moreover, as detailed above the analysis assumptions were un­
realistic. The VC requirements analysis in the literature has focused on extreme
cases where hosts are senders and receivers of all multicast groups, or have focused
solely on pt-mpt VCS, ignoring any pt-pt VC requirements.

To summarise, no analysis has appeared in the literature that investigates the
trade-off between the VC Mesh and MCS approaches, and when each should be
applied. This thesis addresses these issues via a detailed performance analysis in
Chapters 6, 7 and 8.

3.4 Inter-Subnet Multicast Delivery

The previous section described several approaches for providing multicast delivery
within ATM subnets. There are also several ways to transmit multicast Internet
traffic between ATM subnets which are described below and illustrated in Fig­
ure 3.5.

**Single Subnet** Treats the ATM network as one large subnet (Figure 3.5(a)).

**Hop-by-Hop** Subnets are connected by conventional mrouters (Figure 3.5(b)).

**Label Switching Router (LSR)** Subnets are connected by label switching routers
(Figure 3.5(c)).

**NHRP Router** Subnets are connected by NHRP capable routers (i.e. NHSs),
allowing direct VCs to be created Figure 3.5(d)).

3.4.1 Single Subnet Approach

The Single Subnet approach treats the entire network as one Internet subnet (i.e.
there are no routers in the interior of the network) [Arm97c, Smi96, SSRW96].
Figure 3.5: Alternative Inter-Subnet Multicast Delivery Approaches
This means that direct VCs can be created between senders and receivers, regardless of their location on the ATM network. Either the VC Mesh or MCS intra-subnet multicast delivery approaches described in Section 3.3 can be employed to interconnect the senders and receivers. Figure 3.5(a) assumes that the VC Mesh approach is used.

As discussed in [Arm97b], the Single Subnet approach requires a large number of VCs, regardless of whether the MCS or VC Mesh intra-subnet approach is followed. This is because each sender must create a VC to either an MCS or to the receivers (depending on which intra-subnet multicast delivery approach is employed). Moreover, either the senders or the MCS must create pt-mpt VCs with a leaf for every receiver. Hence, the number of receivers that can be supported in the single subnet approach is limited by the UNI 3.1 15 bit leaf node identifier [Arm97b]. However, even with this limitation a single pt-mpt VC can support over 32000 leaves. It is not clear whether future multicast applications would ever involve this number of receivers in a single group. Furthermore the pt-mpt VC leaf requirements of the Single Subnet approach can be reduced by deploying multiple MCSs, or by creating multiple pt-mpt VCs to different subsets of receivers.

Another major disadvantage of the Single Subnet approach is that multicast group membership information must be maintained over the entire network. This has a number of repercussions. Firstly, the load on the signalling network caused by adding a new sender or receiver can be significant. [Arm97b] argues that as the subnet size increases, the rate of group membership changes can grow to the extent that the load on the signalling network is adversely affected. However, no analysis is provided to indicate how large the subnet must be to produce excessive signalling traffic. The second repercussion is that as the size of the subnet (in terms of the number of group members or geographic distance) increases, the time to add a new receiver or sender will also increase. The load on the signalling network can potentially be reduced by staggering when each sender adds a new receiver [Arm97b, Smi96, SSRW96]. However, this will increase the time to add a receiver even further. Moreover, this approach is not applicable when adding new senders.

[Arm97b] discusses many of these issues, however no quantitative analysis has been published that determines how large a subnet must be to make the single subnet approach impractical. Indeed [Arm97b] states "It is hoped that more detailed, quantitative analysis of cluster sizing limits will be prompted by this document".
3. Providing Internet Services over ATM

3.4.2 Hop-by-Hop Approach

The Hop-by-Hop, or MRouter approach, see Figure 3.5(b), attempts to overcome the scalability problems of the Single Subnet approach by dividing the ATM network into subnets connected by mrouters. This approach has two major advantages [Arm97b]. Firstly, data from different senders that is destined for the same next hop can be carried on a single VC, reducing VC requirements. Secondly, multicast group membership information only needs to be maintained on a per-subnet basis. This means that when a new sender or receiver joins a multicast group, only those hosts and routers attached to the same subnet need to modify their VCs. This reduces the signalling resource requirements associated with adding new receivers or senders compared to the Single Subnet approach. Like, the Single Subnet approach, any intra-subnet multicast delivery approach can be used together with the Hop-by-Hop approach. Indeed different subnets may employ different intra-subnet delivery schemes.

No analysis has appeared in the literature to examine, in which network environments, splitting the network into subnets produces significant improvements in resource requirements or group membership addition delay. Moreover, no analysis has been performed to indicate whether there is a significant difference in performance when different intra-subnet approaches are employed. The primary disadvantage of the Hop-by-Hop approach is that, as in the unicast case, multicast traffic must be reassembled at each mrouter it transits. This will increase end-to-end delay compared to the Single Subnet approach. However no literature has appeared that examines whether the end-end delay increase is significant.

3.4.3 Label Switching Router Approach

The Label Switching Router (LSR) approach combines the Hop-by-Hop and Single Subnet approaches. Like the Hop-by-Hop approach, the LSR approach still deploys routers within the network. However rather than reassembling packets at all LSRs, each LSR can map incoming VCs to outgoing VCs. This enables ATM cells to be switched directly through the LSR without datagram reassembly. This removes the reassembly delay problems of the Hop-by-Hop approach. However, as can be seen in Figure 3.5(c) in the extreme where all LSRs directly switch all flows, this approach has the same VC requirements as the Single Subnet approach. This is because cells from different senders, and hence different AAL 5 PDUs, can not be interleaved on the same VC, even when destined for the same set of receivers, unless VC merging is employed (which reintroduces the reassembly delay).

Given the LSR approach still separates subnets with mrouters, multicast group
Existing VC
New Leaf

Figure 3.6: Adding a new Receiver in the Label Switching Approach

information only needs to be maintained on a per-subnet basis [Arm97b]. However, if VC merging is not used, adding a new sender or receiver will trigger the creation of VCs throughout the entire ATM network. This is because the inability to interleave cells requires a separate VC for each sender at each LSR, as shown in Figure 3.5(c). This also means that adding a new receiver requires the local subnet LSR to add the receiver to one VC for each sender, as illustrated in Figure 3.6. These problems can be avoided by reassembling AAL 5 PDUs at the LSRs, via VC merging as discussed in Section 3.2. However, when VC merging is applied, reassembly (or at least buffering of the entire PDU) occurs at each LSR, causing the LSR approach to perform like the Hop-by-Hop approach.

Hence a major difference between the LSR and Single Subnet approaches is that multicast group management can occur on a per subnet basis rather than over the entire subnet. This is because the creation of VCs can be triggered by the flow of data, or Internet multicast routing protocol messages, rather than via group management protocols. Similarly when adding a new receiver, the local LSR is responsible for adding the receiver to VCs rather than the senders or MCS. Hence multicast group management information does not need to propagate across subnet boundaries when the LSR approach is employed.

3.4.4 NHRP Router Approach

The NHRP Router approach also represents a combination of the Single Subnet and Hop-by-Hop inter-subnet delivery approaches. In this approach a multicast routing protocol, e.g. PIM-SM, is used to trigger the creation of direct VCs that bypass interior routers. [RF96] states that reducing the number of routers multicast traffic transits improves performance. However no quantitative analysis is
Figure 3.7: Direct VC Creation using PIM-SM and NHRP

provided to show significant performance gains are made.

[FMR98, RF96] describes an approach where PIM-SM and NHRP are used to create direct VCs. However, this approach is also easily extended to the CBT multicast routing protocol. The PIM-SM/NHRP approach is illustrated in Figure 3.7, where the receiver is attempting to join a multicast forwarding tree, rooted at the sender. In standard PIM-SM, receivers request to join the multicast forwarding tree by sending join messages to their next hop PIM router (e.g. NHS 2 in Figure 3.7). In the NHRP Router approach, the receiver employs NHRP to determine the ATM address of the root of the multicast tree. The PIM join message can then be transmitted directly to this ATM address, rather than to the next hop router (Message 3 in Figure 3.7). When the root of the multicast tree receives the join message, it adds the receiver to its pt-mpt multicast forwarding VC. Once the receiver is successfully added to the direct VC, it can remove itself from the hop-by-hop forwarding tree by sending a PIM-Prune message to its next hop router (Message 5 in Figure 3.7).

Figure 3.7 illustrates the case where both the sender and receiver are attached to the ATM network. The NHRP Router approach is also applicable when either the receivers or the root of the multicast forwarding tree are not attached to the ATM network. In transit ATM networks the direct forwarding VC is created between the ingress router closest to the root of the tree and the egress router(s) closest to the receiver(s).

One disadvantage of the NHRP Router approach is that all leaves will send PIM join messages to the root of the pt-mpt direct VC to refresh PIM soft-state information. However, this can be overcome by using the ATM hard state information (i.e., the fact the leaf is still attached to the VC) to update the PIM soft-state information rather than periodically transmitting PIM messages [RF96].
The key difference between the LSR and NHRP Router approaches is the mechanism used to create the direct ATM VC across the network. The NHRP Router approach uses multicast routing protocol information to create the direct VC. Some proposals employing the LSR approach also propose that multicast routing information trigger the creation of cut-through VCs (e.g. [RDR+97]). However, other LSR based proposals use application based flow classification methods to trigger the direct VC creation. Furthermore, the LSR approach creates the VC in a hop-by-hop fashion. In contrast, the NHRP Router approach creates an end-to-end VC. Hence, the NHRP Router approach suffers the same drawback as the Single Subnet approach where group membership information must be maintained across the entire network.

Rekhter and Farinacci recommend that direct VCs should only be created for applications with QoS requirements, or transmitting large volumes of data [RF96], regardless of whether the LSR or NHRP Router approach is employed. This is due to the high cost associated with creating direct multicast VCs. However, no analysis is provided to indicate (1) the cost associated with creating direct pt-mpt VCs and (2) what volume of application data makes it beneficial to create a direct VC, rather than transmitting data hop-by-hop.

Figure 3.5 illustrates how the VC requirements of the LSR and NHRP Router approaches can be as high as the Single Subnet approach, even though routers are present in the interior of the network (in each segment of the network, two VCs must be supported in the example shown). This is because the routers no longer aggregate multiple traffic streams onto a single VC.

A hybrid approach is also possible where direct VCs are created between interior routers, rather than to the hosts themselves [FMR98]. This will mean that multicast data will be reassembled at the host subnet boundaries, but not within the interior of the network as illustrated in 3.8. This is referred to as the Boundary approach. The Boundary approach can be applied to the Single Subnet, LSR and NHRP Router approaches. Clearly, the end-to-end delay will be higher in the Bound-
Providing Internet Services over ATM

ary variant of these approaches than if ATM VCs are created end-to-end across the entire network. However their delay should be lower than the Hop-by-Hop approach since reassembly only occurs at the edge of the network. Furthermore the resource requirements of the Boundary approach will fall between the direct VC based approaches and the Hop-by-Hop approach. Clearly analysis is necessary to compare all inter-subnet delivery approaches to determine which represents the best solution both in terms of performance and resource requirements.

3.4.5 Performance Analysis Issues

This review has clearly highlighted the need for a performance comparison of alternative inter-subnet multicast delivery approaches. In particular it is not clear whether dividing the ATM network into multiple subnets significantly increases transmission delays, particularly when the LSR or NHRP Router approach is employed. Furthermore the Single Subnet approach is an extension of the intra-subnet approaches where the network is treated as one large subnet. Analysis is needed to determine how large the subnet can become before the resource requirements of this approach are too high. The inter-subnet approaches also differ in the scope of multicast group management information. Analysis is required to determine how each of the approaches compare both when the size of the multicast group, and the frequency of multicast group membership changes vary.

A major motivation of the NHRP Router and LSR approaches is to remove the reassembly delays inherent in the Hop-by-Hop approach. However, this review has shown that analysis is necessary to determine whether end-end delays achieved by the Hop-by-Hop approach are significantly greater than the other approaches. Finally, it is possible to follow the Boundary approach where reassembly occurs only at the edge of the core network. Given the clear trade-off between delay performance and the resource requirements of the alternative inter-subnet approaches, analysis is required to determine whether the Boundary approach provides a good compromise.

3.5 The Multicast Delivery System

The review of inter-subnet approaches presented in this section has focused on the case where each sender creates a separate multicast forwarding tree to receivers, (i.e. where the VC Mesh approach is used). However each of the inter-subnet delivery approaches can be used in conjunction with either the VC Mesh or MCS intra-subnet multicast delivery approaches. Indeed subnets comprising the ATM network may employ different intra-subnet delivery approaches.
When providing Internet multicast delivery over ATM, not only the intra-subnet and inter-subnet approaches need to be considered. A multicast forwarding tree approach must also be selected to interconnect routers in the Hop-by-Hop, LSR and NHRP Router approaches. As described in Section 2.3, Internet multicast routing protocols create either Source or Shared Forwarding Trees. For instance, in PIM-SM Shared Forwarding Trees are created by selecting a rendezvous point (RP) router, to which all senders forward their data. The MCS intra-subnet and Shared Forwarding Tree approaches follow the same principle where senders transmit data to a single device (MCS or RP), which then forwards the traffic to all receivers. Similarly, the VC Mesh and Source Forwarding Tree approaches follow the same principle of forwarding traffic directly from senders to receivers. Clearly the performance of the router based inter-subnet multicast delivery approaches needs to be considered in conjunction with both the Source and Shared Forwarding Tree approaches.

To enable senders and receivers to participate in, or leave, multicast groups, the multicast delivery system must also provide multicast group management mechanisms. The method employed will directly affect the behaviour of multicast delivery systems as senders and receivers join/leave the multicast group. As discussed in Chapter 2, traditionally IGMP is employed to manage multicast groups within subnets, and multicast routing protocols between subnets. However, IGMP is not well suited to ATM networks because it relies on a broadcast capability. MARS [Arm96] is the only currently standardised approach for multicast group management in an ATM subnet that does not rely on broadcast. Each of the delivery approaches described in this section can employ the MARS multicast group management approach within subnets. However, all approaches that divide the ATM network into multiple subnets also require inter-subnet multicast group management to construct the multicast forwarding tree. Given PIM-SM provides multicast group management to create both Source and Shared Forwarding Trees it can be employed for inter-subnet multicast group management.

This section has described alternative approaches for: intra-subnet multicast delivery; inter-subnet multicast delivery; and multicast forwarding trees. This discussion has clearly highlighted the need for quantitative performance analysis of the approaches. In particular, network operators require guidelines or recommendations of when each multicast delivery approach should be applied so that Internet multicast delivery over ATM is provided as efficiently as possible. All current multicast implementations employ the Hop-by-Hop inter-subnet delivery approach. In recent times, researchers have claimed that the end-end delays incurred by this approach are too great and have proposed the alternative Single
Subnet, LSR and NHRP 'direct or Cut-Through' approaches (i.e. approaches that create direct VCs across the network). When supporting static multicast groups these will have the same performance.

The Single Subnet, LSR and NHRP Router approaches can be employed in two modes: (a) across the entire network (as described above), or (b) within the core network only. If the first variant is assumed we term the three approaches as the Cut-Through approach. If the second variation is assumed the three approaches are referred to collectively as the Boundary Approach. Table 3.2 lists all of the alternative multicast delivery approaches described in this chapter.

<table>
<thead>
<tr>
<th>Inter-Subnet Approach</th>
<th>Forwarding Tree Approach</th>
<th>Intra-Subnet Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-Through - Single Subnet</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Cut-Through - LSR</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Cut-Through - NHRP Router</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Cut-Through - Single Subnet</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Cut-Through - LSR</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Cut-Through - NHRP Router</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Hop-by-Hop</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Hop-by-Hop</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Hop-by-Hop</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Hop-by-Hop</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - Single Subnet</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - LSR</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - NHRP Router</td>
<td>Source</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - Single Subnet</td>
<td>Source</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - LSR</td>
<td>Source</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - NHRP Router</td>
<td>Source</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - Single Subnet</td>
<td>Shared</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - LSR</td>
<td>Shared</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - NHRP Router</td>
<td>Shared</td>
<td>VC Mesh</td>
</tr>
<tr>
<td>Boundary - Single Subnet</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - LSR</td>
<td>Shared</td>
<td>MCS</td>
</tr>
<tr>
<td>Boundary - NHRP Router</td>
<td>Shared</td>
<td>MCS</td>
</tr>
</tbody>
</table>

Table 3.2: Alternative Multicast Delivery Approaches

From Table 3.2, it is clear that many multicast delivery approaches have been proposed in the literature. Given this large number of approaches it is very difficult for network operators (and users) to know which multicast delivery approaches best suit their environment. This decision process is made even more difficult by the lack of detailed analysis in the literature that shows (1) the traditional Hop-by-Hop approach is insufficient and (2) the new Cut-Through or Boundary approaches offer significant performance gains. Moreover, more knowledge about the scalability of these approaches is required.
3.6 Conclusions

It is impractical to consider all of the issues for providing Internet services over ATM in a single review. As a result, this chapter has focused on delivery mechanisms for providing unicast and multicast delivery approaches. Many different delivery mechanisms have been proposed, leading to much confusion in industry as to which delivery approaches are best suited to which operating environments. This problem is exacerbated, by the fact that little performance analysis of delivery mechanisms has appeared in the literature. The remainder of this section details the shortcomings in the current literature highlighted by this review.

3.6.1 Unicast Delivery

- No analysis has appeared in the literature that shows that the Cut-Through unicast delivery approach produces significantly lower delays than the Hop-by-Hop approach, particularly when the direct VC must be created. This is even though the development of the Cut-Through approach is based on this premise.

- The VC and signalling resource requirements of the Cut-Through approach are clearly higher than the Hop-by-Hop approach. However, no analysis has been published to determine in which network environments (1) the Hop-by-Hop approach has significantly lower resource requirements, and (2) when the resource requirements of the Cut-Through approach are such that it can not be supported by the network.

- The literature recommends that the Hybrid approach be employed when data arrives if the necessary Cut-Through approach direct VC does not exist. However, no analysis has been performed to show this provides better response time performance than the Buffered or Retransmission approaches.

- The resource requirements of delivery approaches will depend on VC holding times. No analysis has appeared in the literature to determine the sensitivity of unicast IP over ATM delivery approaches to the VC holding time.

- The Label Switching approach is motivated by the fact that the Cut-Through approach takes too long to create direct VCs. However, no analysis has been performed to ascertain (1) whether there is a significant delay associated with creating direct VCs via the Cut-Through approach and (2) whether the Label Switching approach significantly reduces the direct VC creation time.
3. Providing Internet Services over ATM

3.6.2 Multicast Delivery

In terms of multicast delivery there are three key issues that must be considered: the delay performance, VC requirements, and dynamic multicast group support provided by the multicast delivery approaches. Specific issues within these broad areas of investigation are itemised below.

Delay Performance

- No analysis has been performed to determine in which situations there is a significant delay difference between the VC Mesh and MCS intra-subnet multicast delivery approaches.

- No analysis has appeared in the literature that examines the sensitivity of the MCS approach to the location of the MCS.

- No analysis has been published that shows that the Hop-by-Hop inter-subnet delivery approach produces significantly higher delays than the alternative direct VC based inter-subnet approaches.

- No analysis has been published that determines if there is any significant delay performance difference between Shared or Source multicast Forwarding Trees in an ATM environment.

- No analysis has appeared that determines whether the Boundary inter-subnet approaches have significantly higher delays than the Cut-Through approaches.

- No analysis has been performed to determine which of the multicast delivery component choices has the greatest impact on the delay performance of the multicast delivery systems.

VC Requirements

- No analysis has been performed to determine when there is a significant difference in the VC requirements of the VC Mesh and MCS approaches that takes into account both pt-pt and pt-mpt VC requirements.

- Analysis is required to compare the VC requirements of the inter-subnet multicast delivery approaches. In particular there is a need to determine (a) the performance boundaries of each approach, and (b) when there is a significant difference between these approaches.
• The difference between the VC requirements of the Source and Shared Forwarding Tree variants of multicast delivery approaches must be ascertained.

• No analysis has been performed to determine which of the multicast delivery component choices has the greatest impact on the VC requirements of the multicast delivery systems.

Dynamic Multicast Group Support

• The literature indicates that it will take longer to add new senders or receivers to a multicast group when using the VC Mesh approach compared to the MCS approach. However no analysis has been published to determine if there is a significant difference in the time it takes the two approaches to add (or remove) senders or receivers.

• Similarly, no literature exists that shows whether the time to add new senders or receivers significantly differs between the alternative inter-subnet multicast delivery approaches.

• No analysis has been published that quantitatively investigates the difference between Source and Shared Forwarding Trees when supporting dynamic multicast groups.

• No analysis has been published to show if there is any significant difference in the dynamic multicast group support of the Single Subnet, LSR and NHRP Router approaches.

• Analysis is also required to determine the different demands placed on the signalling network when adding (or removing) senders or receivers. The relative signalling network demands of the intra-subnet, inter-subnet and alternative forwarding tree approaches must be analysed.

• No analysis has been performed to determine which of the multicast delivery component choices has the greatest impact on the dynamic group support performance of the multicast delivery systems.

The small number of performance studies of multicast delivery approaches that have appeared in the literature are based on either small artificial network scenarios or assume multicast groups containing a small number of senders. In general analysis is required (in all three areas described above) that employs realistic application traffic characteristics and focuses particularly on the wide area
environment. Much of the multicast delivery analysis that has appeared in the literature has focused on a small portion of the overall multicast delivery system (e.g. intra-subnet delivery approaches). Analysis is required that examines the entire multicast delivery system as a whole since this directly influences the performance users perceive.

From the shortcomings identified above for both unicast and multicast delivery, it is clear that a detailed quantitative analysis of alternative delivery approaches is required. Many approaches have been proposed in the literature, however the relative merits of these approaches have not been determined. This thesis provides quantitative studies to determine the relative performance of the alternative delivery approaches.

Chapter 4 describes the analysis methodology applied in the remainder of this thesis. Chapter 5 then analyses the performance of alternative unicast delivery mechanisms. This is followed in Chapter 6 by a delay performance analysis of alternative approaches for providing Internet multicast delivery over ATM networks. The VC requirements of the multicast delivery approaches are then investigated in Chapter 7. This is followed by a performance study of multicast delivery approaches when supporting dynamic multicast groups in Chapter 8.

The objective of these analyses is to provide network providers with recommendations that enable them to select the delivery mechanisms that best suit their user and network characteristics. This thesis considers delivery mechanisms for unicast and multicast traffic, in terms of delay and resource requirements. The overall findings of this thesis are presented in Chapter 9.
3. Providing Internet Services over ATM
Chapter 4

Methodology to Compare IP over ATM Approaches

4.1 Introduction

Chapter 3 concluded that although many proposals for providing Internet services over ATM have been proposed, little performance analysis of these approaches has occurred. Given the large number of proposals it is difficult for ATM equipment vendors, network providers and users to determine which approaches best meet their service requirements. This dilemma can be seen even now with vendors providing both LANE and CLIP to support unicast best effort Internet services over ATM.

Many proposals state that a given approach is not scalable with respect to a given parameter (e.g. the number of senders in a multicast group). However, no literature has examined the performance boundaries of approaches (i.e. at which parameter values these approaches are no longer applicable). Moreover, the performance of many approaches appears to rely on the values of many interacting parameters (e.g. link capacity, i.e. the bit rate of the link), processor speed, utilisation, number of hops between source and destination). To summarise, without detailed performance comparisons to rely on, it is very difficult to decide which approach for delivering Internet traffic over ATM networks is the best for a given network environment. This thesis addresses this need by recommending which unicast and multicast delivery approaches should be employed to deliver Internet traffic over ATM networks. These recommendations are based upon a detailed performance comparison of approaches via the methodology described in this chapter.

An important aspect of any analysis is determining which performance mea-
4. Methodology to Compare IP over ATM Approaches

sures are important when comparing approaches. Section 4.2 details the performance measures selected to compare approaches for delivering Internet traffic over ATM networks. These are based on the parameters of most interest to network operators and their users. This is followed in Section 4.3 by a description of the features the analysis requires to adequately compare the delivery approaches. Section 4.4 then describes the analysis techniques employed in the remainder of the thesis. This includes a description of how the analysis requirements outlined in Section 4.3 are met.

4.2 Performance Measures

A crucial aspect of the analysis is to determine which performance measures are important when comparing delivery mechanisms for carrying Internet traffic over ATM networks. Moreover, the perspectives of the end-user and the network operator must be considered. This section details the key performance measures for unicast and multicast delivery approaches.

4.2.1 Unicast Performance Measures

The performance measure of greatest importance to users is the latency they perceive. Users are typically concerned with end-end delay, i.e., the time the user must wait until they can use the data they requested. For bulk transfer applications (e.g. WWW, FTP, e-mail) the total response time is important. Total response time is the time until all of the application data requested is received by the user. For interactive or real-time play-back applications (e.g. telnet, video on demand, telephony) the per packet delay is important, since the contents of each datagram can be used as it arrives. Hence, both total response time and per packet delay must be examined when comparing unicast delivery schemes since both types of application must be supported.

Moreover, for some types of application, mean total response times and packet delays are of greatest concern, e.g. for delay insensitive applications. However for real-time, delay sensitive applications delay percentiles are of more interest, particularly for applications with strict end-end delay guarantees such as a telephone call between two users. This is because they give a better indication of the proportion of packets that don't arrive within the delay bounds, and hence impair the quality of the application.

The network providers objective is to meet user performance requirements while maximising revenue. Hence network providers are concerned about resource usage. In an ATM environment the resource requirements to carry the data traffic
4. Methodology to Compare IP over ATM Approaches

(data network requirements) and the requirements to create the necessary VCs (signalling network requirements) must be considered. In terms of the data network, the following performance measures are important:

- Number of VCs required
- Bit rate requirements
- Processing requirements at network nodes (routers, ATM switches etc.).

When comparing the signalling network requirements of approaches the frequency of VC creation or tear-down requests must be considered. This in turn affects the:

- Signalling network link bit rate requirements
- Processing requirements at network nodes (to process the signalling messages)

The relative significance of each parameter will depend on the capabilities of a specific operator's network. For instance, in some networks link capacity is scarce, yet signalling processor resources are plentiful, whereas in other networks the reverse is true. Thus to make general recommendations to operators the performance of approaches must be examined in a variety of network scenarios.

4.2.2 Multicast Performance Measures

Users of multicast applications are also concerned about latency. However, in a multicast application, multiple users receive a copy of the same data. Given the users will be distributed across the network, the delay perceived by different users will differ. Hence, measures such as the time for the data to reach all users and the variation in the latency perceived by different users must be examined.

The performance measures of interest to network providers when supporting multicast delivery are a superset of those described for unicast delivery. Furthermore, because multicast traffic follows multiple paths through the network to reach all of the users, the meaning of the unicast performance measures broadens. For example, the VC, bit rate and processor requirements are distributed across the network, rather than being isolated to one path between the source and destination. Moreover, in multicast applications, both senders (originators of multicast traffic) and receivers (destinations of multicast traffic) can join or leave a multicast group during the life-time of the multicast application. Due to the dynamic nature of multicast groups the following performance measures are also important (in addition to those listed for unicast traffic):
4. Methodology to Compare IP over ATM Approaches

- User Performance Measures
  - The time for data from a new sender to reach all receivers.
  - The time for data from existing senders to reach a new receiver.

- Network Provider Performance Measures
  - The processor requirements to add or remove senders and receivers.

The remainder of this chapter describes an analysis methodology that enables each of these performance measures to be examined for each of the unicast and multicast delivery approaches.

4.3 Analysis Requirements

One observation made in Chapter 3 regarding the small amount of analysis that has been performed, is the unrealistic assumptions that were made. To accurately compare the performance of delivery approaches, it is crucial that they are compared in realistic network and application scenarios. This section outlines the features the analysis must provide to adequately compare alternative approaches for delivering Internet traffic over ATM networks.

4.3.1 Traffic Models

To quantitatively compare alternative delivery approaches, traffic models must accurately reflect the characteristics of traffic produced by Internet applications. Traditionally, most telecommunications network analysis has assumed that traffic arrivals are Poisson. However, several recent studies have shown that Internet traffic is not Poisson, even when many data streams are aggregated together [PF95, DB96, CB96, JBC97]. Indeed [Pax94] shows that the arrival process of different Internet application differs markedly. This has two effects: (1) application specific arrival processes are required, and (2) delivery schemes must be compared when carrying different application traffic. Therefore, the analysis must be sufficiently flexible to model various application arrival processes.

To show the importance of accurately modelling the arrival process Figure 4.1 shows the affect of the arrival process Squared Coefficient of Variation (S.C.V.) \( \frac{\text{mean}^2}{\text{variance}} \) on the time to transmit an ATM cell through an ATM switch, as link utilisation varies. This assumes one ATM switch and an outgoing link with an available link bit rate of 5 Mbits/s. Figure 4.1 clearly shows the arrival S.C.V. has a significant impact on cell transmission delay, particularly when the link is
moderately to heavily utilised. In this example, the link is modelled as a GI/D/1 queue (deterministic service times because ATM cells are fixed length, and a general independent arrival process). There is currently some debate as to whether Internet traffic is self-similar, rather than independent. For instance, [PF95] shows that Poisson processes are valid for modelling the arrival of user sessions, but that wide area packet arrival processes may be better modelled via self-similar processes. [JBC98] shows that HTTP response packet sizes (and hence IP datagram arrival rates) are not infinite, and hence HTTP response traffic may not be self-similar. Furthermore, HTTP request packet arrivals appear to follow a log-normal process [JBC97]. Given this issue is still under debate we assume a general independent arrival distribution for this thesis.

All of the graphs presented in this chapter have been produced by a queuing analysis methodology we have developed based on the Queuing Network Analyser [Whi83]. The analysis methodology is described fully in Section 4.4. In this section it is employed purely to illustrate why each of the analysis requirements is necessary.
4. Methodology to Compare IP over ATM Approaches

4.3.2 Segmentation and Reassembly

As discussed in Section 4.2 bulk transfer, interactive and real-time playback applications must all be supported by IP over ATM delivery schemes. The user is primarily concerned about latency. For many applications (e.g. WWW, e-mail, FTP) this is the total response time rather than the per-packet delay. Hence, the total time for all datagrams comprising the application PDU to arrive at the destination must be calculated. Furthermore, ATM networks employ small fixed size cells as the transportation medium. Thus the IP datagrams must be broken into ATM cells for transmission across the network. This means that even for interactive, or real-time applications the IP datagram must be reconstructed (potentially from thousands of ATM cells) before the application can utilise the data.

Much existing analysis focuses on individual packet delay (i.e. the end-end delay for a single ATM cell or a single IP datagram) [CCL93, OMM91, VHFB89]. Moreover, as stated in [ET90] analytical techniques for single packet delays are well known. However determining the delay for an entire application PDU is more difficult. Given the latency the user actually perceives is based on the arrival of a complete application PDU (which may be comprised of many IP datagrams), the analysis must calculate the response time for entire PDUs. However at the same time the analysis must also capture the fact that the PDU is broken into smaller units (i.e. into one or more IP datagrams which are then broken into ATM cells) for transmission across the network.

Figure 4.2 presents the time for one IP datagram, 9180 bytes in length (the MTU for ATM networks), to be transmitted a distance of 1 km via three ATM switches modelled as single queues (e.g. a campus network). The link bit rate is assumed to be 5 Mbits/s and all ATM switches are assumed to require 10 μs to process each ATM cell [MS95]. The three curves in Figure 4.2 represent: (a) the multiplication of the time to deliver one ATM cell end-to-end and the number of ATM cells comprising the IP datagram (i.e. the worst case where one cell is not delivered until its predecessor reaches the destination); (b) the time to deliver the IP datagram end-to-end when it is transmitted as a single packet rather than being broken into ATM cells; and (c) the time to transmit the IP datagram end-end assuming it is broken into ATM cells and reconstructed at the destination. The objective of this graph is to determine whether it is necessary to model segmentation or reassembly, or whether a simpler delivery model can be employed.

Figure 4.2 shows that methods (a) and (b) overestimate the datagram transmission delay, particularly when the network is heavily utilised. These approaches both over-estimate the delay because they are unable to capture the pipe-lined
4. Methodology to Compare IP over ATM Approaches

Figure 4.2: Effect of Segmentation and Reassembly on Delay Performance
4. Methodology to Compare IP over ATM Approaches

delivery of ATM cells comprising the datagram. The multiplication approach provides the worst estimate because it assumes one cell can not be transmitted until the previous cell has reached the destination. The purely packet level approximation also over-estimates delay. This is because it assumes the datagram travels as one block (i.e. store and forward delivery), where the datagram can not be forwarded from a node until it has completely arrived at that node.

Approach (c) models the segmentation and reassembly of datagrams into cells at the sender and receiver respectively. This captures the pipe-lining affect where the early cells in the datagram can be processed by the first switch while the remainder of the datagram is still being transmitted by the sender. Figure 4.2 clearly shows the importance of modelling the segmentation and reassembly process. The analysis in Figure 4.2 assumes the delivery of a single datagram and an arrival S.C.V. of 1.0. If the arrival S.C.V. increases, or the application PDU is comprised of multiple datagrams, the difference between the approximations increases even further.

This section has shown that any analysis focusing on IP datagram (or application PDU) delay must capture the segmentation and reassembly of IP datagrams. Furthermore, for many applications, particularly real-time applications with strict end-end delay requirements, delay percentiles must be known to adequately compare the relative performance of approaches. Therefore the methodology must be able to calculate both mean delay and delay percentiles.

Another parameter related to the impact of segmentation and reassembly on approach performance is the size of the MTU. In this thesis, we assume an MTU size of 9180 bytes. This size is selected since it is the default MTU size specified for Classical IP over ATM. However, in operational networks smaller MTU sizes may be observed. For instance, many hosts do not have direct ATM connectivity, instead they are connected via other network technologies (e.g. Ethernet) to a router which has ATM connectivity. In this type of environment an MTU of 1500 bytes is usually employed. In addition in some backbone networks MTU sizes of around 500 bytes are employed. A decrease in the size of MTU, will increase the number of IP packets needed to transmit a given volume of application traffic (and hence the overall header overhead). However, the per-packet reassembly delay at each router will decrease. The difference in MTU sizes only becomes an issue when large application responses are involved. Even in this case with the trade-off between the increase in the number of packets and the decrease in per packet reassembly delay, our analysis indicated that a change in MTU size would not have a significant impact on overall approach performance. In this thesis we assume hosts have ATM connectivity and thus we assume an MTU size of 9180 bytes.
4.3.3 Flow Control

The previous section assumed the application PDU comprises one IP datagram. However, for many applications, the application PDU is larger than the IP MTU on ATM networks, hence the application must be broken into multiple IP datagrams. Furthermore, many of these applications (e.g. WWW, FTP) employ TCP as their Internet transport protocol. TCP is a connection oriented, reliable transport protocol that provides flow control and congestion control. As stated in [HOT97] a key aspect of application performance (e.g. the WWW) over TCP is the slow start behaviour.

To simplify the analysis this thesis assumes a lossless network environment. The impact of loss on delivery approach performance is discussed in Section 4.6. The result of the lossless network assumption is that the analysis in this thesis will show the minimum difference in performance between delivery approaches that operators can expect in practice. Assuming a lossless network the only aspect of TCP flow control that must be modelled is the slow start mechanism. TCP transmission is controlled via two windows: the advertised window and the congestion window (cwnd). The advertised window limits the volume of unacknowledged
4. Methodology to Compare IP over ATM Approaches

data present in the network. We assume the window is sufficiently large that it never limits the flow of traffic. If this is not the case, the network link capacity will never be utilised efficiently, since TCP would not allow the application to use all of the available capacity.

The congestion window states the number of unacknowledged TCP segments currently allowed in the network. When the TCP connection is created, only one TCP segment is allowed to be transmitted. Each time the source receives an acknowledgment, the congestion window is increased by one. For example, when the receipt of the first TCP segment is acknowledged, the congestion window is increased to two. Hence two further TCP segments are allowed to be transmitted. Figure 4.3 shows the slow start mechanism in operation. This describes TCP behaviour for older implementations (e.g. 4.3BSD Tahoe). As discussed in [HOT97] modern implementations tend to implement delayed acknowledgments which means that the congestion window will open more slowly than indicated in Figure 4.3. Hence this can be considered to be a best case TCP slow start analysis.

It will take longer for an application PDU to reach the destination when slow start is employed, compared to when the sender is able to transmit all datagrams immediately after one another. Figure 4.4 examines the impact of modelling slow start on total response time assuming two networks: (a) containing three ATM switches and (b) containing three IP routers.

Figure 4.4 shows that when the network contains switches, modelling flow control has little impact on the total response time, regardless of the application PDU size. This is because the dominant delay component is the time to transmit each datagram from the sender. This delay is incurred regardless of whether slow start is employed, because the sender can only transmit one datagram at once. Hence, when sender transmission delay dominates, having to wait for an acknowledgment to arrive has little impact on the total delay. This is because the acknowledgment arrives before the sender has finished transmitting the previous datagrams.

Figure 4.4(b) shows that when routers are employed the total response time is much greater when slow start is modelled, than when it is ignored. In this case the time for each datagram to travel to the receiver (once it has been transmitted from the sender), and the time for the corresponding acknowledgment to arrive start to dominate. The stepping function in the delay graph when slow start is employed occurs because each time an acknowledgment arrives, the congestion window (cwnd), increases by one. The arrival of the acknowledgment also means that one of the previously unacknowledged datagrams has safely arrived at the receiver. Therefore, on receipt of each acknowledgment, two additional datagrams
can be transmitted before the sender must stop and wait for another acknowledgement to arrive.

Figure 4.4 has shown that modelling the TCP slow start mechanism is important for two reasons: (1) response times can be significantly increased by the use of slow start and (2) the impact of slow start on response times will vary for different delivery approaches. Hence, the analysis must accurately model the slow start mechanism. The techniques used to produce the results in Figure 4.4 are described in Section 4.4.3.

4.3.4 Multiple Flows

In Section 4.3.1 it was shown that the application traffic characteristics can have a significant effect on response time performance. In practice many applications (e.g. the WWW) are comprised of a large number of message flows with different characteristics. For example, when a user 'clicks' on a hyper-link, this causes: (1) a TCP connection to be created, (2) a WWW request to be transmitted, (3) WWW response packets to be returned, and (4) acknowledgments to be generated for the WWW response packets. The analysis methodology must be able to capture the characteristics of the different flows and the interaction between them. Furthermore, the total response time calculation must account for the delay contribution of all of the message flows triggered by the application request.

4.3.5 Accounting for Signalling Traffic

As discussed in Chapter 3 one of the key differences between delivery approaches is their VC and hence signalling network requirements. The analysis must be able to capture the delay associated with creating VCs as well as the delay for transmitting data across the network. The analysis must also enable the load placed on signalling processors to be determined.

4.3.6 Accurately Modelling Routers and Switches

Another key difference between alternative unicast and multicast delivery approaches is whether they deploy IP routers or ATM switches. The two key differences between these devices are: (1) routers must reassemble datagrams before processing them whereas switches process individual cells; and (2) traditionally routers are only able to process one datagram at a time, whereas switches can process multiple cells in parallel. The analysis must model both routers and switches in a manner that captures the differences between them.
4. Methodology to Compare IP over ATM Approaches

Figure 4.4: Effect of TCP Slow Start on Application PDU Delay

(a) Intermediate ATM Switches

(b) Intermediate IP Routers

Figure 4.4: Effect of TCP Slow Start on Application PDU Delay
Figure 4.5 compares the difference in the response times achieved when the network comprises of switches and routers. The analysis assumes a stream of MTU sized (9180 byte) IP datagrams and that the network comprises four network nodes (routers or switches). One can see from Figure 4.5, that response times are significantly greater when routers are employed. Given the different delay characteristics of the two devices and that different approaches deploy routers and switches differently, it is crucial that the analysis models routers and switches accurately.

It is important to note that this thesis assumes a traditional router architecture. In recent times new router architectures have been proposed which makes them more like ATM switches. If such routers are employed in a network, the performance of approaches will be better than indicated in this thesis. Hence, in this sense this is a worst case analysis.

4.3.7 Modelling Multicast Flows

As discussed in Chapter 3, multicast delivery approaches employ pt-mpt ATM VCs. When a pt-mpt VC is created between a sender and several receivers, only
one copy of each cell is transmitted on any link between the senders and receivers. When the path to two receivers diverges, the ATM switch duplicates the cell and transmits a copy of the cell via each of the outgoing interfaces. Given several multicast delivery approaches (e.g. the VC Mesh and MCS intra-subnet approaches) use pt-pt and pt-mpt VCs differently it is critical that the analysis models pt-mpt message flows accurately.

Figure 4.6 shows the response time performance of transmitting traffic to three receivers using either three pt-pt VCs or one pt-mpt VC. The network is assumed to comprise of two switches with the sender attached to one switch and the three receivers to the second switch. The performance of the two approaches are compared as the arrival rate is varied to produce the indicated utilisation on the links connecting the receivers to the second switch.

This figure indicates a significant difference in the delay performance of the two approaches. Indeed when the pt-pt modelling approach is employed, a far greater volume of traffic must flow in the network causing it to overload when the receiver link utilisation exceeds 0.3. This is because the sender must transmit one copy of each cell for each receiver. This analysis has shown that it is not feasible to model pt-mpt VCs via several pt-pt VCs. Hence the analysis must provide a mechanism to model the forwarding characteristics of pt-mpt VCs.

4.3.8 Scalability

A key requirement of the analysis methodology is scalability. The analysis must be able to model realistic wide area network scenarios, comprising large distances and network nodes. Furthermore, the analysis must be able to model many simultaneous message flows each potentially having different traffic characteristics.

4.4 Analysis Methodology Description

Telecommunications network performance is commonly analysed via queuing network analysis or simulation. One potential danger with simulations is that when the simulation output data is not analysed correctly, erroneous results may be produced. Firstly, transient period detection is required to ensure data is only collected from the steady state phase of the simulation. Secondly, for standard statistical measures such as variance to be produced, the simulation results must be independent and identically distributed (i.i.d.). This means simulation techniques such as Independent Replications (which runs the same simulation multiple times with each run producing one observation) and Batch Means (where sequences of simulation observations are grouped into batches, each producing one observa-
4. Methodology to Compare IP over ATM Approaches

Figure 4.6: Comparison of Pt-Pt and Pt-Mpt VC Performance
4. Methodology to Compare IP over ATM Approaches

In contrast, queuing analysis methods, when used in conjunction with accurate arrival and service models, can produce meaningful results from one iteration instead of lengthy iterations of a simulation. As a result, larger systems are often modelled via queuing analysis rather than simulation. This thesis considers networks containing hundreds of queues and traffic flows. With simulation techniques and the processing power available, it was infeasible to use simulation to obtain results for networks of this size.

The main criticism of queuing analysis is the number of simplifying assumptions that must be made to produce a solution. A class of queuing analysis techniques addresses this criticism by performing an approximate analysis of an exact model instead of an exact analysis of a simplified model. Several such packages have been developed including the Queuing Network Analyser (QNA) [Whi83] and QNET [HN90] which analyse networks of GI/G/m queues.

QNA employs a parametric decomposition method where nodes are analysed separately once the internal flow parameters are calculated. In contrast, QNET replaces the queuing network with an approximating Brownian system model. The stationary distribution of this model is then computed by solving a highly structured partial differential equation problem. As stated in [HN90], although QNET was found to be generally more accurate than the original version of QNA it requires much more extensive computation. An improved version of QNA has also been developed [Whi94] which computes the departure process from each node on a per class basis rather than as an aggregate. This allows the arrival processes of internal flows to be modelled more accurately. As stated in [FS92, Whi94], determining aggregated departure flows can lead to inaccurate approximations when: (a) flow service times differ, (b) some flows have very low or high arrival process variability, or (c) when the routing of flows is deterministic. Hence we believe when this improvement is incorporated into QNA the accuracy of the approximations are sufficient. Refer to [Whi94] for a discussion of the accuracy of this extension compared to simulation and other parametric decomposition approximations.

To apply the per flow departure process approximation to QNA we derived the per flow arrival process S.C.V. to each queue, based on the per flow departure approximations in [Whi94] (see Equation 4.1). This extends Equations 24 to 26 of [Whi83] which calculate the aggregate arrival S.C.V. to each queue. Equation 4.1 expresses $c^2_{ijk}$, the arrival S.C.V. of flow $k$ at node $j$, in terms of three constants $a_{jk}, d_{ijk},$ and $e_{ijkl}$, where $i$ refers to the source node, $j$ the next hop
4. Methodology to Compare IP over ATM Approaches

node, \( k \) to the flow of interest and \( l \) to all other flows. \( \omega_{jk} \) describes the superposition operation for flow \( k \) at node \( j \). All other quantities are described in Table 4.1. The aggregate traffic arrival S.C.V. can then be described by Equation 4.2. The remainder of this section details how the QNA based analysis meets the analysis requirements outlined in Section 4.3.

\[
c_{ajk}^2 = a_{jk} + \sum_{i=1}^{n} (d_{ijk} \times c_{aik}^2) + \sum_{i=1}^{n} \sum_{l=1, l \neq k}^{r} (e_{ijkl} \times c_{ail}^2) \tag{4.1}
\]

where

\[
a_{jk} = 1 + \omega_{jk} \times ((p_{0jk} \times c_{0jk}^2) - 1) + \sum_{i=1}^{n} p_{ijk} \times (q_{ijk}(\rho_{ik}^2 \times c_{sik}^2 + t_{ik} \times \sum_{l=1, l \neq k}^{r} \rho_{il}^2 \times t_{il}^{-1} \times c_{silt}^2) + (1 - q_{ijk})),
\]

\[
d_{ijk} = \omega_{jk} \times p_{ijk} \times q_{ijk} \times (1 - 2 \times \rho_{ik} \times \rho_{i} \times \rho_{ik}^2),
\]

\[
e_{ijkl} = \omega_{jk} \times p_{ijk} \times q_{ijk} \times t_{ik} \times \rho_{il}^2 \times t_{il}^{-1},
\]

\[
\omega_{jk} = \frac{1}{1 + 4 \times (1 - \rho_{jk})^2 \times ((\sum_{i=0}^{n} P_{ijk}^2)^{-1} - 1)}
\]

\[
1 - \omega_{jk} = \omega_{j} \times \sum_{k=1}^{r} \frac{c_{aik}^2}{\lambda_{jk}} \tag{4.2}
\]

where

\[
\omega_{j} = \frac{1}{1 + 4 \times (1 - \rho_{j})^2 \times (v_{j} - 1)}
\]

\[
v_{j} = \frac{1}{\sum_{k=1}^{r} \frac{\lambda_{jk}}{\sum_{l=1}^{r} \lambda_{jl}}}
\]
### 4. Methodology to Compare IP over ATM Approaches

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(k)</td>
<td>flow of interest</td>
</tr>
<tr>
<td>(l)</td>
<td>all other flows</td>
</tr>
<tr>
<td>(j)</td>
<td>the node of interest</td>
</tr>
<tr>
<td>(i)</td>
<td>the previous node</td>
</tr>
<tr>
<td>(n)</td>
<td>number of nodes</td>
</tr>
<tr>
<td>(r)</td>
<td>number of message flows</td>
</tr>
<tr>
<td>(\lambda_{jk})</td>
<td>mean arrival rate of flow (k) at node (j)</td>
</tr>
<tr>
<td>(c_{a_{jk}}^2)</td>
<td>S.C.V. of flow (k) arrivals at node (j)</td>
</tr>
<tr>
<td>(c_{a_j}^2)</td>
<td>aggregate S.C.V. of arrivals at node (j)</td>
</tr>
<tr>
<td>(r_{jk})</td>
<td>mean service time of flow (k) at node (j)</td>
</tr>
<tr>
<td>(c_{s_{jk}}^2)</td>
<td>S.C.V. of flow (k) service time at node (j)</td>
</tr>
<tr>
<td>(p_{ijk})</td>
<td>the proportion of arrivals of flow (k) to node (j) that came from node (i)</td>
</tr>
<tr>
<td>(q_{ijk})</td>
<td>the proportion of flow (k) messages that go to node (j) from node (i)</td>
</tr>
<tr>
<td>(t_{jk})</td>
<td>the proportion of all departures from node (j) that are flow (k)</td>
</tr>
<tr>
<td>(\rho_j)</td>
<td>the utilisation at node (j)</td>
</tr>
<tr>
<td>(\rho_{jk})</td>
<td>flow (k)'s contribution to the utilisation at node (j)</td>
</tr>
<tr>
<td>(n_k)</td>
<td>the number of nodes visited by flow (k)</td>
</tr>
<tr>
<td>(n_{kj})</td>
<td>the (j)th node visited by flow (k)</td>
</tr>
<tr>
<td>(\gamma_i)</td>
<td>the multiplicative factor for all flows leaving node (i)</td>
</tr>
<tr>
<td>(\gamma_{ik})</td>
<td>the multiplicative factor for flow (k) when leaving node (i)</td>
</tr>
<tr>
<td>(c_{o_{jk}}^2)</td>
<td>the s.c.v. of the external flow (k) arrival process to node (j)</td>
</tr>
<tr>
<td>(p_{o_{jk}})</td>
<td>the proportion of external arrivals to node (j) that are flow (k)</td>
</tr>
<tr>
<td>(EW_{jk})</td>
<td>the expected waiting time for flow (k) at node (j)</td>
</tr>
<tr>
<td>(ET_k)</td>
<td>total sojourn time for flow (k)</td>
</tr>
<tr>
<td>(Var_k)</td>
<td>variance of the total sojourn time for flow (k)</td>
</tr>
</tbody>
</table>

Table 4.1: QNA Parameter Definitions
4. Methodology to Compare IP over ATM Approaches

4.4.1 QNA Traffic Modelling

Figure 4.1 showed the importance of accurately modelling traffic arrival process characteristics. QNA describes the arrival process for each flow via a two moment approximation: (a) the mean arrival rate, and (b) the arrival process S.C.V. [Whi83]. This allows the burstiness of application flows to be captured. Moreover, QNA requires the mean service time and service time S.C.V. to be specified for each flow at each queue visited. Hence QNA is able to capture both the service and arrival characteristics of application flows.

4.4.2 Modelling Segmentation and Reassembly

Section 4.3.2 highlighted the importance of modelling the segmentation and reassembly of IP datagrams into cells, as the traffic traverses the network. QNA provides a customer combination and creation facility where a multiplicative factor, $\gamma_i$ can be applied to all departures from a given queue $i$ (see Section 2.2, [Whi83]). There is customer creation if $\gamma_i > 1$ and customer combination if $\gamma_i < 1$. This could be employed to model the segmentation and reassembly of IP datagrams into ATM cells (or application PDUs into datagrams).

The problem with the approach described above is that the characteristics of flows may differ significantly (e.g. the size of IP datagrams and hence the number of cells it is comprised). For instance a datagram carrying a TCP acknowledgment should be broken into 2 cells, and a 9180 byte datagram into 192 cells. Hence the existing QNA customer creation/combination facility is not directly applicable. To overcome this deficiency, we extend QNA to allow the multiplicative factor to vary for different flows at the same queue. This is achieved by modifying the QNA input by flows and routes to include $\gamma_{ik}$, the multiplicative factor for flow $k$ at queue $i$. As a result, the flow rate equations (Equations (4) and (5) from [Whi83]) are expanded to include $\gamma_{ik}$ as shown in Equations 4.3 and 4.4, where $1H$ is the indicator function of the set $H$, i.e., $1H(x) = 1$ if $x \in H$ and $1H(x) = 0$ otherwise (see Section 2.3, [Whi83]). Furthermore, now $\gamma_{jk}$ is incorporated into the flow rate equations, $\gamma_j$ is no longer required in the following QNA equations: (18),(22),(23),(26),(43),(73) and (74) in [Whi83].

\begin{equation}
\lambda_{ijk} = \sum_{l=1}^{n_k-1} \lambda_{ik} \gamma_{ik} 1(k,l) \cdot n_{kl} = i, n_{k,l+1} = j \quad (4.3)
\end{equation}

\begin{equation}
\lambda_{i0k} = \lambda_{ik} \gamma_{ik} 1k \cdot n_{k} = i \quad (4.4)
\end{equation}
Figure 4.7 illustrates how $\gamma_{ik}$ is employed to model segmentation and reassembly. The network is assumed to comprise of a sender, one router and two switches. Two flows are assumed: (a) an MTU size IP datagram flow (192 cells) and (b) an acknowledgment flow (2 cells). The sender and router process entire IP datagrams, whereas the switches process individual ATM cells. Assuming the overall arrival rate $\lambda$, of both flows is 1.0 datagram/s, Figure 4.7 shows that the arrival rate of both flows to the sender and router is $\lambda_{21} = 192$, whereas the arrival rate of the flows to each switch (e.g. $\lambda_{21}$ and $\lambda_{22}$) is multiplied by 192 and 2 respectively. This is because the IP datagrams comprising the two flows segment into 192 and 2 cells respectively. Likewise, when the flows move from the first switch to the router, $\gamma_{21}$ and $\gamma_{22}$ are 1/192 and 1/2 respectively. This has the effect of combining the cells to create a flow of datagrams at the router.

4.4.3 Calculating the Impact of Flow Control on Delay

As shown in Section 4.3.3, modelling TCP flow control can have a significant impact on the delay calculated by the analysis. Therefore, response time calculations for applications employing TCP, must account for this staggered delivery of TCP segments. [HOT97] shows that if delayed acknowledgements are not employed, $2^i$ TCP segments can be sent when the $i$th acknowledgement has been received. Figure 4.8 shows the pseudo code we developed to determine the response time for an application PDU comprised of $k$ MTU sized segments. In practice, an application PDU will comprise of $k - 1$ MTU sized segments and one smaller segment. In addition, the delivery of the segments across the network will be pipelined. For example, the third TCP segment will be sent immediately after the second TCP segment. Thus, even if the final TCP segment is significantly smaller than the MTU, the delay observed by that final segment will be influenced by the delay observed by the previous larger segments. In the best case, the previous segment will have left each network element before the final segment reaches that network.
function tcp_delay

input:
    mtu_delay - time to deliver MTU segment to receiver
    ack_delay - time to deliver ACK from receiver to sender
    trans_delay - time for sender to transmit MTU segment
    k - number of segments in the application PDU

output:
    delay - time for application PDU to arrive at receiver

if k is odd
    if k = 1
        delay = mtu_delay
    else
        delay = trans_delay +
            tcp_delay(mtu_delay, ack_delay, trans_delay, k - 1)
    endif
else
    delay = 0
    for i = 1 to k/2
        possible_delay = mtu_delay + ((mtu_delay + ack_delay) * i) + (trans_delay * (k - i*2))
        if possible_delay > delay
            delay = possible_delay
        endif
    done
endif

Figure 4.8: Algorithm to Determine TCP Response Time

element. In this scenario the delay to deliver the previous segment will not affect the final small segment. In the worst case, the final segment will be transmitted immediately after the previous segment, and hence the delay observed by the final segment will be similar to the previous segment (since the final segment must queue behind the previous segment across the entire network). In this thesis we assume the worst case where all of the segments (even the final potentially small segment) experience the same delay.

To explain Figure 4.8 consider an application PDU comprising 4 segments. Expanding the for loop of the algorithm in Figure 4.8, one can see that the response time for the application PDU is given by:
max

\[ mtu\_delay + (mtu\_delay + ack\_delay) + (trans\_delay \times 2), \]
\[ mtu\_delay + ((mtu\_delay + ack\_delay) \times 2) \]

(4.5)

As shown in Figure 4.3, the fourth segment can be transmitted when two acknowledgments have been received by the sender. This is expressed by the second component of Equation 4.5. However in circumstances where the transmission time is the dominant component of the MTU delay, the sender may still be transmitting the third segment when the acknowledgment for the second segment arrives. In this case the fourth segment can not be transmitted until the sender finishes transmitting the third TCP segment. This is expressed by the first component of Equation 4.5. QNA will produce the mtu\_delay, ack\_delay and trans\_delay figures by solving the queuing network as described in the following section.

4.4.4 Modelling Multiple Flows and Total Response Time

Section 4.3.4 discussed the importance of being able to capture the characteristics and interactions of different message flows. QNA allows two types of input: one for a single aggregate type of traffic and the other input via classes and routes [Whi83]. In the latter case the arrival process for each class is described independently. Furthermore for each class the route it takes through the queuing network must be specified along with its service process at each queue visited. A separate class can be employed for each message flow, enabling QNA to capture the characteristics of each message flow, and how the message flows interact with one another (e.g. when two flows queue at the same queuing network node).

As discussed in Section 4.2, users are often interested in total response time (also known as sojourn time) rather than per packet latency. Hence it is crucial that the analysis methodology is able to calculate total response times. Note the total response time (e.g. the time to download a WWW page) may be comprised of multiple flows (e.g. a request flow and a response flow). Thus the delay contribution of each flow must be included. Once the internal flow parameters have been determined, QNA treats each queue comprising the network model as a GI/G/m queue. The steady state waiting time is then calculated for each queue as described in Section 5 of [Whi83]. The mean and variance of both the total waiting time and sojourn time is calculated as described in Section 6.3 of [Whi83]. The expected total sojourn time and variance of the total sojourn time for flow k are shown in Equations 4.6 and 4.7 respectively.
4. Methodology to Compare IP over ATM Approaches

\[ ET_k = \sum_{j=1}^{n_k} (\tau_{kj} + EW_{kj}) \]  \hspace{1cm} (4.6)

\[ Var_k = \sum_{j=1}^{n_k} (\tau_{kj}^2 c_{skj}^2 + Var(W_{kj})) \]  \hspace{1cm} (4.7)

As discussed in Section 4.3, when the user makes an application request (e.g. clicks on a WWW hyper-link), this triggers several message flows. For example, in the case of WWW traffic: (a) the creation of a TCP connection, (b) the transmission of a WWW request, and (c) the receipt of the response. Hence to determine the total response time mean and variance, the delay contribution of each flow must be included. Equations 4.8 and 4.9, show the overall sojourn time calculations, assuming r flows comprise the data transfer.

\[ ET = \sum_{k=1}^{r} ET_k \]  \hspace{1cm} (4.8)

\[ Var = \sum_{k=1}^{r} Var_k \]  \hspace{1cm} (4.9)

In addition to mean waiting time and variance, the waiting time distribution is often important, particularly for real-time applications. This is because waiting time percentiles give greater information about the number of packets that are likely to arrive within the delay constraints of real-time applications (e.g. a voice or video conference) Any packets arriving after the strict end-end delay deadline can not be used by the application.

QNA approximates the waiting time distribution by fitting a hyper-exponential, exponential or Erlang-2 distribution to the first two moments of the waiting time (see Equations (55) to (61) in [Whi83]). The appropriate distribution is selected based on the value of the s.c.v. of the conditional delay that the server is busy. This thesis employs an improved version of the waiting time distribution fitting procedure developed in [Law94]. In this procedure an r-stage Erlang distribution is used instead of the 2-stage Erlang distribution when the conditional delay s.c.v. is less than 0.501. R, the number of stages is given by the reciprocal of the conditional delay s.c.v. This approximation provides greater accuracy, however there is no closed form expression for the cumulative distribution function of an r-stage Erlang distribution. Therefore the probability density function has...
to be integrated numerically, which is infeasible for large values of r. However, as shown in [Kle75], as r increases, the r-stage Erlang distribution tends to the normal or Gaussian distribution due to the central limit theorem. In this thesis, if the number of stages r, is greater than 50, the waiting time distribution is fitted to a normal distribution. For smaller values of r, the r-stage Erlang distribution is employed.

4.4.5 Incorporating Signalling Network Performance

In ATM networks, signalling VCs and data VCs are created. Signalling VCs are often CBR, hence they can be considered in isolation from the data VCs. In this thesis, two queuing networks are created for each telecommunications network. One is used to model the transmission of signalling traffic and the other data traffic. If delivering an application PDU requires the creation of a VC (or any other signalling traffic), the end-end delays for the signalling flows involved are added to the end-end delays for the data flows producing the overall response time. This method also enables us to analyse the affect of network parameters on signalling network and data network performance independently, allowing performance boundary and robustness studies to specific parameters to be undertaken.

4.4.6 Modelling Switch and Router Behaviour

The key difference between many delivery schemes is the manner in which they deploy routers and switches. As shown in Section 4.3.6, the performance of switches and routers can differ significantly. This is for two reasons: (a) routers must reassemble IP datagrams whereas switches do not and (b) routers can only route one IP datagram at once whereas a switch can forward several cells through the switching fabric simultaneously. In this thesis the first aspect is captured by employing the segmentation and reassembly mechanism described in Section 4.4.2. At routers, customers are combined to create a stream of IP datagrams, however at switches the message flow is treated as a stream of ATM cells. The second difference between routers and switches is captured in the queuing model employed. Routers are modelled as a series of input and output queues connected by a single processor queue. In contrast the switch is modelled by a series of input, switching fabric and output queues. The queuing models for each of the network nodes considered in this thesis are described further in Section 4.5.
4. Methodology to Compare IP over ATM Approaches

4.4.7 QNA Scalability

The major focus of this thesis is the performance analysis of delivery mechanisms for Internet traffic over ATM WANs. As a result it is crucial that the analysis methodology can scale to large networks. One benefit of queuing analysis is that multiple replications are not required to produce statistically valid results. However the time to solve each queuing network must be reasonable. In QNA the time to solve a queuing network depends on the number of network nodes and message flows.

Figure 4.9 shows the CPU time to solve the queuing network as the number of network nodes and message flows vary. This assumes a 233MHz Pentium Processor and 64 Mbytes RAM. QNA is implemented as a Matlab program. This experiment showed that QNA tended to be memory limited. This is because QNA must solve a system of linear equations, whose size depends on the number of network nodes and message flows [Whi83].

From Figure 4.9, we see that as either the number of message flows or network nodes increases, the CPU time to solve the queuing network increases. When there is one message flow and 1000 network nodes, it takes approximately 25 minutes to
solve the queuing network. When the number of message flows increases to ten, it takes approximately 45 minutes to solve a queuing network with 500 nodes. As the number of message flows increases further to fifty, networks with fifty nodes can be solved also in around 45 minutes. Therefore, it is feasible to employ QNA when modelling large networks and applications that generate a large number of message flows. Note the vertical lines when modelling one or five flows indicate regions where the time to solve the queuing network was too small for the Matlab CPU timer to measure.

4.5 Queuing Network Models

In this thesis, we assume end-to-end connectivity is possible between hosts. Thus to employ queuing analysis, queuing network models are required for senders, receivers, MARS, MCSs, switches, routers and LSRs. Figure 4.10 shows the queuing model assumed for each type of network node. As can be seen from Figure 4.10, senders, receivers, MARS and MCSs are modelled by a receiver, processor and transmitter queue. The receiver queue is present solely to allow segmentation and reassembly between datagrams and cells to be modelled. The service time at this queue is zero. The transmitter queue models the transmission of a datagram into the network, and hence the service process depends on the size of the datagram and the link rate. The processor queue service process depends on the specific network node and the message flow involved.

Figure 4.10 also shows the queuing model assumed for a router. As discussed in Section 4.4.6, the router is modelled as a series of receiver and transmitter queues connected by a single processor queue. Again the receiver queues are present only to model segmentation and reassembly. All datagrams have to be processed by the central processor queue before being forwarded to the appropriate transmitter queue for transmission to the next network node.

ATM switches are modelled by a series of input, switching fabric and output queues. The input queues are present to allow the routing of cells from the input queue to the appropriate output queue to be modelled (their service time is assumed to be zero). That is, we assume the switches employ output queuing rather than input queuing. Figure 4.10 also shows we assume each switch has a single signalling processor associated with it. All signalling traffic must pass through the switching fabric twice, once to reach the signalling processor, and once to be switched to the appropriate output queue.

Chapter 3 also described delivery schemes that employ devices that can act as both a switch and a router (e.g. a Label Switching Router). These nodes
Figure 4.10: Queuing Network Models
allow some traffic to be switched at the ATM layer, while others are routed at the IP layer. Figure 4.10 shows how this hybrid switch and router is modelled. The VPI/VCI is used to identify which cells must be routed. These cells are forwarded to the router receiver queue for reassembly into datagrams for routing. Once the routing decision is made, the datagram is segmented into cells at the router transmitter queue which are forwarded through the switching fabric to the appropriate switch output port. Cells which do not require IP layer processing pass directly from the input port through the switching fabric to the output port as in a typical ATM switch.

4.6 Conclusions

This chapter has detailed the analysis methodology employed in this thesis. Firstly the performance measures the analysis must produce to adequately compare the relative performance of unicast and multicast delivery schemes were described. To accurately compare the alternative delivery schemes it is also crucial that their performance is examined in realistic network and application scenarios. Section 4.3 described the features the analysis requires to accurately compare the alternative delivery approaches. This showed the analysis must:

- Characterise the burstiness of the arrival process.
- Model the segmentation and reassembly of application PDUs, IP datagrams and ATM cells as the traffic flows across the network.
- Account for the TCP slow start mechanism when calculating response times.
- Calculate the total response time for an application PDU, accounting for all of the message flows required to deliver the application PDU to its destination.
- Model the performance of both data traffic and signalling traffic.
- Accurately model the different behaviour of switches and routers.
- Model the flow of traffic over point-to-multipoint VCs.
- Scale to large WANs carrying large numbers of message flows

Section 4.4 described the queuing analysis methodology we have developed based on QNA which meets each of these performance requirements. In the remainder of this thesis this analysis methodology and the queuing models in Sec-
4. Methodology to Compare IP over ATM Approaches

4.5 are employed to compare the performance of schemes for delivering unicast and multicast Internet traffic over ATM networks.

In terms of loss the literature has clearly shown the importance of mechanisms such as Early Packet Discard (EPD) on switch performance, particularly when carrying TCP traffic [RF95, LE96]. If switches employ EPD, then the relative performance of approaches (regardless of whether they employ routers or switches) when carrying UDP traffic will be similar regardless of the loss rate. This is because UDP does not have a retransmission mechanism, and EPD will ensure that the volume of additional traffic due to loss will be no higher in the switch based network than the router based network. Note this assumes there is equal likelihood of loss at routers and switches. The study of relative router and switch buffer size requirements to meet a given loss rate is an area of research in its own right and is not considered further here.

When TCP traffic is being carried, the relative difference in the overall delay performance of approaches is sensitive to end-end delay and TCP packet size. This is because, given the same loss rate at switches and routers, all approaches will generate the same number of TCP retransmissions, and each retransmission incurs another end-end delay. Hence the higher the loss rate, the greater the impact of the end-end delay difference of approaches on their overall relative performance. In this thesis we assume a zero-loss network. Thus the analysis presented throughout this thesis shows the minimum delay difference expected between approaches. In practice (i.e. in networks where loss does occur) the delay difference between approaches will be higher. It is also important to note that the size of the TCP packets being transmitted will impact the likelihood of loss. Hence the larger the TCP packets being transmitted the greater the difference between the delay figures shown in this thesis and those operators would expect to see in a live network. Thus the analysis presented in this thesis should be considered as a boundary case that shows the minimum difference in performance operators can expect between approaches. In practice, the importance of selecting an approach with lower end-end delays will be even greater than indicated by the analysis in this thesis.
4. Methodology to Compare IP over ATM Approaches
Chapter 5

Unicast Delivery

5.1 Introduction

In Chapter 3 the Hop-by-Hop, Cut-Through and Label Switching approaches for unicast delivery of Internet applications over ATM were introduced. It was shown that all currently available solutions for providing Internet services over ATM use one or more of these approaches to provide unicast delivery.

The relative merits of these alternative unicast delivery mechanisms has been the subject of much debate (refer to the IETF Internetworking Over NBMA (ION) Working Group Archives), however no detailed quantitative analysis of these approaches has appeared in the literature. Moreover most protocol developments and comparison efforts have focused on a LAN environment. However, ATM is increasingly being deployed, and indeed was designed as a WAN technology. Hence efficient mechanisms are required to provide unicast delivery over ATM WANs. This chapter provides a detailed quantitative analysis of approaches for providing Internet unicast delivery over ATM WANs. As outlined in Chapter 3 the major issues that must be examined are to:

- Determine whether the Cut-Through unicast delivery approach produces significantly lower delays than the Hop-by-Hop approach, particularly when the direct VC must be created.

- Determine in which network environments (if any) the VC and signalling resource requirements of the Cut-Through approach are such that they cannot be supported by the network.

- Investigate whether the Hybrid approach provides better response time performance than the Buffered or Retransmission approaches.
- Analyse the sensitivity of unicast IP over ATM delivery approaches to the VC holding time.

- Determine whether there is a significant delay associated with creating direct VCs via the Cut-Through approach and from this whether the Label Switching approach significantly reduces the direct VC creation time.

Much of the growth of the Internet that has occurred in recent years is due to the introduction of the World Wide Web (WWW) [BCGP92]. The WWW is currently the most popular Internet application, both in terms of the number of active WWW sessions, and the volume of traffic transmitted. Hence in the majority of this chapter, performance comparisons of the unicast delivery mechanisms assume WWW traffic. As discussed in Chapter 4, the performance of approaches are sensitive to application characteristics. Hence the performance of unicast delivery approaches when carrying smaller volumes of connection-oriented or connection-less application traffic is also considered.

In Section 5.2 the analysis methodology is outlined. This includes a description of the WWW protocol, its traffic characteristics and the network model used to compare the unicast delivery approaches. The performance of the unicast delivery mechanisms are compared in Section 5.3. The analysis compares the response time performance of the approaches in a variety of operating scenarios. The bit rate and signalling network resource requirements of each approach are also examined assuming a range of VC holding times. This chapter concludes that the Cut-Through approach produces significantly lower response times than the Hop-by-Hop approach even when the direct VC must be created. Furthermore, the Buffered approach should be used to create direct VCs. However, in networks where there is insufficient signalling capacity to create direct VCs in a timely fashion, the Hop-by-Hop approach should be employed. The analysis also shows that the resource requirements of moderately sized WWW proxies can easily be supported by current ATM network equipment.

5.2 Analysis Methodology

The WWW is currently the most popular Internet application requiring unicast delivery. Hence the majority of this chapter compares unicast delivery approaches when carrying WWW traffic. Initially an overview of the WWW application is presented. Using this description and WWW traffic traces, an application and network model of the WWW is developed. These models are used in conjunction with
the analysis methodology developed in Chapter 4, to compare the performance of the unicast delivery approaches.

### 5.2.1 WWW Overview

A typical network structure for WWW access is shown in Figure 5.1. Users (WWW clients) navigate the WWW by following hyperlinks [BCGP92]. Some requests can be serviced directly by the WWW client using an internal cache. However, most requests must be forwarded to a WWW server. The WWW server returns one or more objects which are parsed by the client and displayed appropriately. The third, and optional, WWW component is the WWW proxy server. These are placed between clients and servers as part of IP firewalls and provide a cache of responses. A proxy accepts requests from clients, forwards them to servers and relays the response to the appropriate client(s).

WWW traffic is carried over the Internet using the Hypertext Transfer Protocol (HTTP) [BLFF96, FGM+97] running over TCP/IP. When transmitting HTTP requests and responses across the network, a number of message flows occur as shown in Figure 5.2. For more information about TCP connection creation and termination see [Ste94].

WWW users are most concerned about response time (i.e. the time between issuing a request and viewing the response as shown in Figure 5.2). Clients, proxies, servers and the network all contribute to WWW response time. In this chapter only the response time performance of the core network is considered. This is because this analysis focuses on the performance of unicast delivery approaches.
Figure 5.2: WWW Message Flows
5. Unicast Delivery

over ATM WANs. [HOT97] has also considered HTTP response time performance over a variety of network technologies. However this analysis did not examine performance over ATM networks. Moreover, it focused on transport protocol layer performance rather than how the traffic was delivered across the network. The objective of [HOT97] was to analyse WWW performance. In contrast, the objective of this chapter is to compare alternative ways to deliver unicast traffic over ATM, and employ the WWW application as an example of unicast traffic.

5.2.2 WWW Traffic Model

To determine the WWW response times produced by the unicast delivery approaches via QNA, each message flow in Figure 5.2 must be modelled via the first two moments of its arrival and service process. The arrival process for each message flow was determined using actual WWW traffic data sets and statistics\(^1\). To determine HTTP request sizes, all WWW traffic carried on a portion of the University of Wollongong backbone was collected. Over twelve weeks (from the 3rd July 1995), 700,000 requests were captured using the Ethernet trace software snoop [Sun]. The remaining traffic statistics and traces were determined from the access log files of the University of Wollongong campus wide proxy. Over 28 weeks (from the 27th November 1995), 7 million requests were logged by the proxy serving 2058 unique client machines.

The mean size and variance of HTTP request and response object sizes are shown in Table 5.1. The response object size does not include the HTTP header which is assumed to be 150 bytes in length.

<table>
<thead>
<tr>
<th>HTTP Data Unit</th>
<th>Mean Size (bytes)</th>
<th>Variance (bytes(^2))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>247</td>
<td>8649</td>
</tr>
<tr>
<td>Response Object</td>
<td>11880</td>
<td>2.19e10</td>
</tr>
</tbody>
</table>

Table 5.1: HTTP Response and Request Size Statistics

As seen in Table 5.1, HTTP responses are typically larger than the IP MTU used in ATM networks (9140 bytes excluding TCP/IP headers). As discussed in Chapter 4.4, application PDUs larger than the MTU payload, (in this case HTTP responses), are modelled as a flow of one or more MTU sized packets followed by a smaller packet. Using the University of Wollongong WWW data sets, the mean and variance of the number and size of MTU and smaller sized datagrams produced by each HTTP response are measured. These statistics are shown in

\(^1\)Many thanks to John Judge, The Telecommunications and Information Technology Research Institute, for providing these data sets and statistics
5. Unicast Delivery

Tables 5.2 and 5.3.

<table>
<thead>
<tr>
<th>HTTP Response Datagram Type</th>
<th>Mean Number of Datagrams per Response</th>
<th>Variance in the Number of Datagrams</th>
<th>S.C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU Sized Datagram</td>
<td>1.02</td>
<td>261.05</td>
<td>251.0</td>
</tr>
<tr>
<td>Small Datagram</td>
<td>1.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Table 5.2: HTTP Response Datagram Count Statistics

<table>
<thead>
<tr>
<th>HTTP Response Datagram Type</th>
<th>Mean Datagram Size (bytes)</th>
<th>Variance in the Size of Datagrams</th>
<th>S.C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU Sized Datagram</td>
<td>9140</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>Small Datagram</td>
<td>2718.4</td>
<td>6.13e6</td>
<td>0.83</td>
</tr>
</tbody>
</table>

Table 5.3: HTTP Response Datagram Size Statistics

Message Flow Arrival Statistics

The statistics in Table 5.2 indicate that on average, each HTTP response consists of 1.02 MTU sized IP datagrams and 1.0 smaller IP datagram. To represent this information in the queuing analysis, assuming an average arrival rate of HTTP responses is \( \lambda \), the average arrival rate of MTU sized datagrams and smaller datagrams is 1.02 \( \lambda \) and \( \lambda \), respectively. The remainder of this chapter assumes the arrival rate, \( \lambda \), is one request per second.

In [JBC97] it was shown that the S.C.V. of HTTP request arrivals and hence arrivals of HTTP responses (since each HTTP request generates one response) is 1.7. Table 5.2 indicates that the S.C.V. in the number of MTU and smaller datagrams generated by each HTTP response is 251.0 and 0.0 respectively.

The arrival statistics of TCP acknowledgments (ACKs) must also be determined. ACKs from the proxy to the server are generated by SYN messages, MTU response packets and small response packets. Furthermore, the analysis assumes that each packet containing part of the HTTP response generates a separate ACK message (acknowledgment) at the proxy. The overall arrival process of ACKs can be determined via superposition of the flows generating ACKs as described in Equation 4.2. This gives a mean arrival rate of 3.02 ACKs from the proxy to the server with an S.C.V. of 87.0. The arrival process of ACKs from the server to the proxy is the same as the TCP connection FIN flow, since ACKs for other flows in this direction can be piggy-backed onto SYN or HTTP response packets.

Table 5.4 summarises the arrival statistics assumed for each message flow in
5. Unicast Delivery

Figure 5.2. Similarly, the message size statistics assumed in the analysis are shown in Table 5.5. This includes AAL 5 and LLC/SNAP protocol overhead when required [Hei93]. When employing the Cut-Through approach, VCs must be created. However we assume that the VCs required by the Hop-by-Hop approach already exist. This is because it is much more likely that VCs created for aggregate Hop-by-Hop traffic will be available than a direct VC to be used solely between a given sender and receiver. The analysis assumes that in the Cut-Through approach the arrival of a HTTP request will trigger the signalling request for a direct VC to the WWW server. The signalling message size statistics were calculated from the ATM Forum UNI 3.1 specification [The94].

### Table 5.4: Message Flow Arrival Statistics

<table>
<thead>
<tr>
<th>Message Flow</th>
<th>Flow Direction</th>
<th>( \lambda ) (messages/s)</th>
<th>S.C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP SYN</td>
<td>Proxy → Server</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>TCP SYN</td>
<td>Server → Proxy</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>HTTP Request</td>
<td>Proxy → Server</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>MTU Datagram Flow</td>
<td>Server → Proxy</td>
<td>1.02</td>
<td>251</td>
</tr>
<tr>
<td>Small Datagram Flow</td>
<td>Server → Proxy</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>ACK</td>
<td>Proxy → Server</td>
<td>3.02</td>
<td>87.1</td>
</tr>
<tr>
<td>FIN</td>
<td>Server → Proxy</td>
<td>1.0</td>
<td>1.7</td>
</tr>
<tr>
<td>ACK</td>
<td>Proxy → Server</td>
<td>1.0</td>
<td>1.7</td>
</tr>
</tbody>
</table>

### Table 5.5: Message Size Statistics

<table>
<thead>
<tr>
<th>Message Flow</th>
<th>Size (bytes)</th>
<th>S.C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>TCP SYN</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>TCP ACK</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>TCP FIN</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>HTTP Request</td>
<td>303</td>
<td>0.14</td>
</tr>
<tr>
<td>HTTP MTU Response Datagram</td>
<td>9196</td>
<td>0</td>
</tr>
<tr>
<td>HTTP Small Response Datagram</td>
<td>2774.4</td>
<td>0.83</td>
</tr>
<tr>
<td>Signalling SETUP</td>
<td>120</td>
<td>0</td>
</tr>
<tr>
<td>Signalling CONNECT</td>
<td>44</td>
<td>0</td>
</tr>
<tr>
<td>Signalling CONNECT ACK</td>
<td>17</td>
<td>0</td>
</tr>
</tbody>
</table>

5.2.3 Network Model

The WWW response times produced by each unicast delivery approach will depend on the distance and the number of hops between the WWW proxy and server. To develop realistic network models, the locations and number of hops to servers logged in the WWW traffic data sets were determined. Table 5.6 summarises the
breakdown of server locations obtained from the proxy log traffic trace. Servers with invalid (or unresolvable) domain names were marked as unknown. The mean number of hops between the 100 most popular servers in each region and the University of Wollongong proxy are also shown in Table 5.6. The number of hops to a given server was determined using Traceroute, which records the route packets follow between two hosts. These figures give an overall mean number of hops of 13.5. This finding is similar to other investigations which have determined the mean number of hops between a source and destination to be 15.7 [Nat96].

<table>
<thead>
<tr>
<th>Location</th>
<th>Percentage of WWW Servers</th>
<th>Mean Number of Hops</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>19.4%</td>
<td>9</td>
</tr>
<tr>
<td>U.S.A</td>
<td>67.2%</td>
<td>14</td>
</tr>
<tr>
<td>Elsewhere</td>
<td>13.0%</td>
<td>18</td>
</tr>
<tr>
<td>Unknown</td>
<td>0.5%</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5.6: WWW Server Locations

Using these statistics, three network scenarios were constructed to compare the unicast delivery approaches. These three scenarios correspond to locating the WWW server in each of the regions described in Table 5.6. Although the number of hops to each region is easily calculated via traceroute, it is difficult to determine the distance travelled [Nat96]. The Distance Service [Onl], was used to determine the distance between two locations. The Australian server was assumed to be located in Melbourne, 800 km from the proxy. The U.S.A. server was assumed to be located in Bloomington, Indiana (e.g. www.yahoo.com), with data travelling from Wollongong via Sydney and San Francisco, giving a total distance of 13,000 km. Finally, the Elsewhere server was assumed to be located in France, 23,000 km from the proxy travelling via Sydney, San Francisco and New York. The analysis approximates realistic lengths for each hop via the Distance Service (e.g. an 80 km link between Wollongong and Sydney).

When employing the Hop-by-Hop approach, each hop between the WWW proxy and server is assumed to be a router. When the Cut-Through approach is examined, each hop is assumed to be an ATM switch. In practice there are likely to be ATM switches between the hops modelled in this chapter. However, since these ATM switches are present, regardless of the unicast delivery approach, they will affect all unicast delivery approaches equally, and hence are not modelled. The queueing network model described in Section 4.5 is used to represent hosts, routers and ATM switches.
5.2.4 Further Analysis Assumptions

As stated in Chapter 4, the characteristics of the data VCs depend on the application being modelled. In this chapter, data VCs are assumed to have a link rate of 5 Mbits/s and to use the UBR service category. Both the ABR and UBR service categories are appropriate for WWW traffic and recently there has been much debate about which approach performs better. Since UBR is the most commonly employed service class for carrying Internet traffic it is assumed in this chapter.

Table 5.7 contains the default transmission and processing delays assumed in the analysis. Some of these default parameter values are varied in the analysis. The Hop-by-Hop performance analysis presented in this chapter is a best case analysis since it assumes there is no segmentation and reassembly processing delay at the router. Furthermore the forwarding delay at routers and switches is assumed to be the same, whereas most current routers are one to two orders of magnitude slower than switches in practice [Cro97].

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation Delay</td>
<td>5 μs/km</td>
</tr>
<tr>
<td>Data VCs</td>
<td>UBR 5 Mbits/s</td>
</tr>
<tr>
<td>Signalling VCs</td>
<td>CBR 1 Mbits/s</td>
</tr>
<tr>
<td>Switch Processing Delay</td>
<td>10 μs</td>
</tr>
<tr>
<td>Router Processing Delay</td>
<td>10 μs</td>
</tr>
<tr>
<td>Signalling Processor Speed</td>
<td>100 VCs/s</td>
</tr>
</tbody>
</table>

Table 5.7: Queuing Analysis Assumptions

To vary the data and signalling network utilisation, background traffic is assumed to be present on both signalling and data networks. On the signalling network, background traffic is modelled as VC creation messages flowing in both directions across the network. For the data network ideally the background traffic would model aggregate Internet traffic. However, no such model exists in the current literature. As discussed in [Pax94], it is very difficult to determine such a model because of the variety of Internet applications. Given HTTP traffic is the highest contributor to Internet traffic, in this analysis the background traffic is also assumed to be HTTP traffic, with the statistics described above.

5.2.5 Unicast Delivery Performance Measures

This chapter compares the performance of unicast delivery approaches from both user and network provider perspectives. This section describes the performance measures used in the remainder of the chapter.
User Performance Requirements

As discussed in Section 4.2, WWW users are most concerned about response time (the time between the user selecting a hyperlink until the response is completely received). Total WWW response time is determined by summing the end-end delay of the following message flows (see Figure 5.2):

- SYN from proxy to server
- SYN from server to proxy
- HTTP Request from proxy to server
- HTTP Response from server to proxy (calculated as described in Chapter 4.4).

To determine when there is a significant difference in the response time performances of approaches, the required level of service must first be known. As mentioned earlier, the primary performance measure of interest to users is response time. Shneiderman [Shn84] investigates user response time expectations. Users establish response time expectations based on several factors including past experience, the perceived complexity of the problem and their tolerance to delays [Mil68, Shn84].

In general users prefer response times of less than one second for interactive applications [Shn84]. However, as users became more experienced, experiments have found that users will expect response times well below one second if this is technically feasible [Shn84].

The user-computer interaction described in [Mil68] and [Shn84] both predate the WWW. However, the WWW has similarities to older interactive applications upon which these papers were based. In [VF95] the response time behaviour of the WWW itself is examined. This paper introduces the concept of the ideal WWW performance, termed the 100-100 Web. The 100-100 Web refers to an operating scenario where servers are available 100% of the time and there is 100 ms latency between selecting a hyperlink and the user receiving the information that link represents. 100 ms is chosen as the ideal delay target because any delay less than one 100 ms is perceived as instantaneous [VF95]. Viles and French investigated the latency of the current WWW from a site in the U.S.A to 550 servers located in the U.S.A. or Europe. They found that the median latency to U.S.A. and European servers were 0.27 seconds and 1.27 seconds respectively. However, these times represent a lower bound on total response time because they only include the time to create the TCP connection and transmit the HTTP request to the
server. They do not include server processing delays or the time to retrieve the requested information.

In the analysis presented in the remainder of this chapter, response times are determined which include the time to receive the entire HTTP response. This is a better performance measure from a users perspective because it is closer to the delay they actually experience. The analysis does not include any processing delays at the proxy or server. This is a separate research issue and affects all unicast delivery schemes equally. However, given server processing delays can represent a significant portion of the total response times [AdM96, AAY97], it is even more important that the response time delay contributed by the network is minimised.

To determine whether the difference in the response time performance of approaches is significant, one must also determine when users start to perceive a difference in delays. Miller found that the variation in delay that a user can detect depends on the magnitude of delay [Mil68]. For delays between 0.6 and 0.8 seconds, users could detect a difference in delay if the delay is 5% more or less than the delay of interest. That is users have a 5% tolerance for delays in the range of 0.6 - 0.8 seconds. As the delay magnitude increases, user tolerance also increases. For delays in the range of 2.0 to 4.0 seconds and for delays greater than 4.0 seconds the delay variation tolerance increased to 8% and 15% respectively [Mil68]. The trend in these results indicate that for delays closer to 100 ms the tolerance will be at most 5% and may be even lower. In the remainder of the chapter these tolerance measures are used to assess the significance of response time performance differences between the unicast delivery approaches.

Network Provider Performance Requirements

A key objective for network providers is to maximise profit. This can often be achieved by minimising the resources required to support a single application. Hence an investigation of the resource requirements of the alternative unicast delivery approaches is essential.

All of the alternative unicast delivery approaches will have similar bit rate requirements since they all transmit the same volume of traffic. As discussed in Chapter 3, the approaches differ in their VC and signalling resource requirements. The volume of both of these resources will depend on the VC holding time (the time the VC remains open after activity on that VC ceases). Hence this chapter compares the VC and signalling resource requirements of approaches for a range of VC holding times.
5.3 Performance Analysis

In this section the performance of alternative unicast delivery approaches are compared in terms of total response time, signalling network requirements and VC requirements. In Section 5.2 three network scenarios were described which vary the WWW server location between Australia, the U.S.A and Europe. The approaches are compared in all of these network scenarios over a range of network parameter values.

As discussed in Chapter 3, the Cut-Through approach requires a direct VC to be created between the sender and receiver, if one does not already exist. Note the Label Switching and Cut-Through approaches will provide the same response time performance if the VC exists. In the remainder of this chapter we refer to the case where the direct VC exists as the Cut-Through approach.

Chapter 3 discussed three ways to handle traffic that arrives before the direct VC is active. The Retransmission approach drops packets that arrive before the VC is active. The Buffered approach buffers the traffic until the VC is active. In contrast, the Hybrid approach sends traffic via the hop-by-hop route, switching to the direct VC when it is created. The only difference between the Retransmission and Buffered approaches is whether packets are buffered or dropped. Buffering is preferable since it means the packets are ready to be sent as soon as the direct VC is open. Hence in the delay analysis we only consider the Buffered and Hybrid approaches. The Retransmission approach should only be used in place of the Buffered approach if buffer capacity is an issue. This is discussed further in the next section.

If the Label Switching approach employs UNI 3.1 signalling to create each portion of the direct VC, response time performance will be similar to the Hybrid Approach. If a lightweight signalling protocol is employed (e.g. the Ipsilon Flow Management Protocol (IFMP) [NEH+96a]), the time to create the direct VC will be reduced. Other Label Switching approaches (e.g. Tag Switching or Ipsofacto) do not have any VC creation delays, since the VCs are either set up in advance based on topology information, or as the data is forwarded through the network. In this case they will have the same performance as the Cut-Through approach where the direct VC already exists. Given the large number of Label Switching approaches that have been proposed in the literature they are not explicitly considered in this Chapter. However the performance of many of the schemes will be the same as the Cut-Through or Hybrid approaches. An area of future research is to investigate the performance of alternative Label Switching signalling techniques in detail.
5. Unicast Delivery

5.3.1 Effect of the Available Bit Rate

The frequency of segmentation and reassembly is the key difference between the Hop-by-Hop, Cut-Through, Hybrid and Buffered approaches. In the Hop-by-Hop approach (and the first phase of the Hybrid approach), IP datagrams must be reassembled at every router. In contrast, in the Cut-Through and Buffered approaches, datagrams are always only reassembled at the edge of the network. The main component of reassembly delay is the time that it takes an entire IP datagram to arrive at the router. This is directly affected by the transmission rate of the link. Hence, initially the relative performance of approaches is investigated as the transmission rate (or available bit rate) varies.

Figure 5.3 shows the mean WWW response time produced by each approach as the available bit rate varies between 1 Mbits/s and 150 Mbits/s. In these results, the data network link utilisation is assumed to be 0.7 for all link bit rates. The signalling network is assumed to have no background traffic and the signalling processors are assumed to process 100 VCs/s.

In all scenarios considered, the Hop-by-Hop approach always produces the highest response times, regardless of the available bit rate. Even when the available bit rate is 150 Mbits/s, the Hop-by-Hop approach response time is twice that of the Cut-Through approach.

The majority of current WANs operate between 2 Mbits/s and 45 Mbits/s (e.g. the Telstra Internet). In this operating region, the Cut-Through approach produces significantly lower response times than the Hop-by-Hop approach. This finding holds, even if the direct VC must be created, using either the Hybrid or Buffered approaches. For example, when the WWW server is located in the U.S.A. the Cut-Through approach response time is seventeen and four times lower than for the Hop-by-Hop approach for link rates of 2 Mbits/s and 50 Mbits/s respectively. In these operating regions the magnitude of delay is on the order of seconds and hundreds of milliseconds for 2 Mbits/s and 50 Mbits/s links respectively. Hence the difference in the response time performance of the approaches is significant.

In the future, WAN link rates are likely to increase. At high link rates it is still better to use the Cut-Through approach in terms of response time. However, the difference in response time performance decreases with increasing available bit rate. When the WWW server is located in the U.S.A. or France, the difference in response times reduces to a factor of two. Again, in this region the magnitude of response times are on the order of hundreds of milliseconds. Even, when it is assumed that the total response time is several seconds once server processing delays are included, the difference in the response time performance of the Cut-Through and Hop-by-Hop approaches is significant, even when the direct VC must
Figure 5.3: Mean WWW Response Time as Data Network Link Bit Rate Varies
be created.

In practice, it is unlikely that 150 Mbits/s transmission rates will be available to individual Internet applications, even when 150 Mbits/s links or greater are deployed in WANs. The reason for this is that ATM is designed to divide the available link bit rate among multiple applications, or classes of application. Thus, even in the future, the Cut-Through approach is likely to provide significant response time savings over the Hop-by-Hop approach. In the remainder of the chapter the available bit rate for the application under investigation is assumed to be 5 Mbits/s.

When examining the effect of the WWW server location, one observes that the relative difference in performance of approaches is similar when the server is located in the U.S.A. or France. However, when the server is located in Australia, increasing the available bit rate has less effect on the relative difference in performance of approaches than the other two scenarios. The only difference between the Australian scenario compared to the U.S.A. and France scenarios is the ratio of the total distance travelled to the number of hops. The effect of this ratio on the relative difference in performance of the unicast delivery approaches is considered further in Section 5.3.4.

Intuitively, one would expect the Hybrid approach to produce lower response times than the Buffered approach. This reasoning is based on the fact that the Hybrid approach allows data to be transmitted before the direct VC exists. However, Figure 5.3 indicates that this is almost always not the case. When the available bit rate is less than 50 Mbits/s, the Buffered approach produces lower response times than the Hybrid approach, regardless of the WWW server location. This occurs due to the combination of two factors: (1) when the available bit rate is low, it takes significantly longer to transmit data via the hop-by-hop path than the direct path; and (2) direct VC creation delay is small compared to the overall WWW response time. As the available bit rate increases, these factors no longer hold, and the Hybrid approach begins to outperform the Buffered approach.

The region where the Buffered and Hybrid approach performance significantly differs depends on the location of the server. For the Australian server the difference is significant at or below 25 Mbits/s. For the U.S.A. and French servers, the performance difference is only significant at bit rates up to 20 Mbits/s. The effect of the location of the server is considered further in Section 5.3.4.

The results in Figure 5.3 also highlight the sensitivity of the Hybrid approach to the delay to create the direct VC compared to the time to transmit each IP datagram hop-by-hop. To explain further, the Hybrid approach can only change to the direct VC when it has a datagram ready to send. For example, when the
Figure 5.4: Sensitivity of the Hybrid Approach to the Transmission Rate
proxy receives the TCP SYN message it checks if the direct VC exists and if so, forwards the HTTP request on the direct VC. Otherwise the hop-by-hop route must be followed. If the direct VC is created just after the HTTP request has been transmitted, the HTTP request will travel the entire route to the destination hop-by-hop. This is shown in Figure 5.4 where the HTTP response arrives earlier when link rate 1 is available instead of link rate 2, even though rate 2 is higher. This is because the SYN message in the link rate 2 case arrives earlier due to the higher transmission rate. This causes the request message to be carried via the hop-by-hop path. In contrast at the slower link rate 1, both the request and response are carried via the direct VC.

This phenomenon can also be seen in Figure 5.3 where the response time performance of the Hybrid approach suddenly increases before decreasing again. In the Australian server scenario, when the available bit rate is less than 50 Mbits/s the hop-by-hop route is sufficiently slow compared to the time to create the direct VC that the TCP SYN and HTTP request message are sent hop-by-hop and the response and acknowledgment packets are sent via the direct VC. However, at 75 Mbits/s, the time to transmit data hop-by-hop has reduced to the extent that the initial response packet is also sent via the hop-by-hop route. This packet takes much longer to transmit hop-by-hop than via the direct path and hence the response time increases. The hop-by-hop route delay then continues to decrease due to the increasing link bit rate, in turn reducing the time that the initial response packet takes to travel hop-by-hop. At 150 Mbits/s the delay has reduced to the extent that the acknowledgment message for the initial response also travels hop-by-hop, increasing the Hybrid approach total response time. The primary advantage of the Buffered approach is that the direct VC can be used as soon as it is created. This means that the Buffered approach is less sensitive to the time to transmit data between the proxy and server, than the Hybrid approach.

A clear disadvantage of the Buffered approach is that traffic must be stored at the proxy until the direct VC is created. In contrast, the Hybrid approach allows traffic to be transmitted immediately. However this is not a major disadvantage because many Internet applications, including the WWW, follow an asymmetric client-server architecture where most traffic is transmitted from the server to the client. This is because the client, or proxy, only transmits the small TCP connection and HTTP request messages, representing a small buffering penalty. The large response packets are not retrieved and buffered by the server until the direct VC is open and the request has reached the server. The only difference between the Retransmission and Buffered approach is whether packets are dropped or buffered if they arrive before the direct VC is created. Given the low volume of packets
that must be buffered it seems better to employ the Buffered approach. This has the additional advantage that the packets can be sent as soon as the direct VC is created, rather than having to wait for the arrival of the re-transmitted packet.

The variance and 95th percentile of WWW response times were also examined as the available bit rate increases. Figure 5.5 shows the variability in the WWW response times produced by the approaches assuming the WWW server is located in Australia. Similar results were obtained when the server was located in the U.S.A. or France.

These results indicate that the difference in the response time variance of the approaches is insensitive to the available link bit rate, except for the Hybrid approach. The Hybrid approach response time variance approaches the Hop-by-Hop variance at high link bit rate. This occurs because most of the WWW traffic travels via the hop-by-hop route rather than the direct route in this region. The 95th percentile response time performance of the unicast delivery approaches follows the same trends as the mean WWW response time and hence is not considered further.

To summarise, this initial investigation indicates that the Cut-Through ap-
5. Unicast Delivery

Unicast Delivery should be used for all link bit rates considered. If a direct VC must be created, the analysis indicates that the Buffered approach should be used when the available bit rate is less than 20 to 25 Mbits/s, depending on the location of the server. However when the available bit rate is greater than this, there is no significant difference in the response time performance of the Hybrid and Buffered approaches. Hence, the Buffered approach is preferred because in regions where the Buffered and Hybrid approach performance differs, the Buffered approach delay performance is better.

5.3.2 Effect of the Data Network Utilisation

In the previous section the effect of available bit rate was investigated, assuming an overall link utilisation of 0.7. This section tests the sensitivity of approaches to the data network link utilisation.

In Figure 5.6 the mean WWW response times produced by the approaches are compared as the data link utilisation varies (note that this variation is caused by varying the volume of background traffic. The link bit rate remains constant at 5 Mbits/s). These results show that the WWW response times achieved by the Hop-by-Hop and Hybrid approaches increase more rapidly with increasing data network utilisation than for the Cut-Through and Buffered approaches. This is because the hop-by-hop path has higher queuing delays than the direct VC path since entire datagrams must be queued at intermediate nodes rather than individual ATM cells.

To illustrate this further, consider the U.S.A. located server. In this scenario, the difference in the Hop-by-Hop and Cut-Through approach response times are a factor of 2, 12 and 25 for data network link utilisations of 0.2, 0.6 and 0.9 respectively. Similar results are obtained for the Australian and French servers. Hence it is clear that the data network link utilisation significantly affects the relative performance of the unicast delivery approaches.

When network utilisation is low, the Hop-by-Hop and Hybrid approaches perform similarly. This is because in this region most packets are sent hop-by-hop before the direct VC is created. Indeed, when network utilisation is low, network resources are wasted by creating a direct VC since it won't be used.

In terms of user perceived performance, the significance of the difference between the approaches at a range of data network utilisations must be considered. The results in Figure 5.6 show that the user perceived difference between the Cut-Through and Hop-by-Hop approaches is significant for all network loads for all server locations if the direct VC exists. When the direct VC must be created there is no significant difference between the Hop-by-Hop, Hybrid or Buffered ap-
Figure 5.6: Mean WWW Response Time as Data Network Utilisation Varies
5. Unicast Delivery

approaches when the network is lightly loaded (up to 0.2 link utilisation). In practice, networks will operate at a moderate to high link utilisation so that network providers can maximise their revenue. Furthermore, given the adaptive nature of IP traffic (particularly TCP based traffic), the IP traffic will use the available link bit rate [Ste96]. In moderately to highly utilised operating regions, significant response time performance gains are provided by the Cut-Through approaches (Cut-Through, Hybrid and Buffered) compared to the Hop-by-Hop approach, regardless of whether a direct VC must be created.

The relative performance of the Buffered and Hybrid approaches also depends on the data network link utilisation. Figure 5.6 shows that as the data network utilisation increases, the Buffered approach performs increasingly like the Cut-Through approach. This is because the VC creation delay becomes less dominant in the overall WWW response time as the data link utilisation increases. Similarly, one would also expect the Hybrid approach to perform increasingly like the Cut-Through approach as the VC creation delay becomes less dominant. However, this is not the case because the time until the Hybrid approach can switch to the direct VC also depends on the time it takes to send data via the hop-by-hop path. As the network utilisation increases it takes longer for the initial TCP connection packets to travel between the proxy and server. Hence it takes longer to change to the direct VC, even though the VC itself is created quickly, compared to the overall response time. Hence, one of the key disadvantages of the Hybrid approach compared to the Buffered approach is that it is highly sensitive to the network utilisation (i.e. the time to transmit data via the hop-by-hop route).

To date, the analysis has assumed that all network links are equally loaded. In operational networks, there are likely to be several highly loaded links, while other links are lightly utilised. In Figure 5.7 the performance of the approaches are compared when the number of heavily loaded links is varied. All other links are assumed to have a utilisation of 0.1. These results show that varying the number of heavily loaded links does not affect the relative performance of unicast delivery approaches. This can be seen clearly in Figure 5.7 by observing where the dip in the Hybrid approach response time occurs as the number of heavily utilised links varies. This indicates that varying the number of heavily loaded links shifts the graphs along the utilisation axis. That is, varying the number of heavily utilised links has the same affect on performance as varying the utilisation across all links. Thus, in the remainder of the analysis, the load on all links is assumed to be equal.

The sensitivity of approaches to the router processor speed was also investigated. This analysis indicated that increasing the router speed to greater than 100,000 packets/second has no affect on the response time performance of the
5. Unicast Delivery

Figure 5.7: Mean WWW Response Time as the Number of Heavily Loaded Links Varies

(a) One Link Heavily Utilised

(b) Half of the Links Heavily Utilised

(c) All Links Heavily Utilised
5. Unicast Delivery

Hop-by-Hop and Hybrid approaches (the only unicast delivery approaches that employ routers). This is because the dominant component of router delay in this region, is the time for the datagram to completely arrive at the router. Reducing the router processor speed below 100,000 packets/second does start to increase Hop-by-Hop and Hybrid approach response times noticeably. However, given most current routers can process at least 100,000 packets/s this situation is unlikely.

5.3.3 Effect of the Signalling Network Characteristics

The comparison of approaches to date has assumed that the signalling processor speed is 100 VCs/s and that there is no background traffic on the signalling network. Furthermore, the analysis has shown that the response time performance of the Hybrid and Buffered approaches is highly dependent on the time to create the direct VC. The two parameters which affect VC creation time are the signalling processor delay and the utilisation of the signalling network. This section analyses the effect of these parameters on the performance of the Hybrid and Buffered approaches.

The relative performance of the Hybrid and Buffered approaches are compared assuming the server is located in the U.S.A. Similar results were obtained for the other server locations. The Hop-by-Hop and Cut-Through response times are included in the results as references, since they are unaffected by signalling network performance. Figure 5.8 shows the response time performance of the approaches assuming data network utilisations of 0.3 and 0.9 as the signalling network utilisation and signalling processor capacity vary.

The results highlight the key disadvantage of the Buffered approach. If the signalling network is highly utilised, the VC creation delay will be large and potentially infinite. Hence the HTTP response time can also be extremely high and potentially infinite, regardless of the data network utilisation. In contrast, the Hybrid approach will never perform worse than the Hop-by-Hop approach. This is because all traffic will be transmitted via the hop-by-hop path if the direct VC does not open. The disadvantage of the Hybrid approach when the signalling network is highly utilised is that signalling network resources are wasted by attempting to create a direct VC that is not used.

Two observations can be made about the performance of the Buffered and Hybrid approaches. The Buffered approach is more robust to high data network utilisations than the Hybrid approach. However, the Hybrid approach is more robust to high signalling network (processor or link) utilisations than the Buffered approach. Network providers can use these observations to select the approach that best suits their network operating conditions (e.g. if the signalling network
Figure 5.8: Mean Response Time as Network Utilisation and Processor Speed Varies

(a) Data Network Utilisation of 0.3

(b) Data Network Utilisation of 0.9
5. Unicast Delivery

tends to be highly utilised, but there is spare data network capacity the Hybrid approach should be selected).

Figure 5.8 also indicates that the response time performance of the Hybrid and Buffered approaches is insensitive to the signalling processor speed, unless the signalling network processor is almost overloading. Hence, the response time performance of the Hybrid and Buffered approaches is unaffected by signalling processor speed, as long as it is sufficient to support the expected volume of signalling traffic. Likewise, the Hybrid and Buffered approaches are insensitive to the signalling network utilisation. This is because the time to create the VC is small compared to the overall response time even when the signalling network is highly utilised.

This analysis has indicated that if the signalling processor speed and signalling network bit rate is sufficient to support the expected volume of signalling messages both the Hybrid and Buffered approaches perform well in terms of WWW response time. Furthermore, if the VC creation delay is small compared to the overall response time, the Buffered approach should be employed. However, if the signalling network processor utilisation is greater than 0.95, the Hybrid approach should be used. Indeed, the Hop-by-Hop approach would be preferred in this region because it does not waste signalling resources attempting to create a direct VC.

Current high end ATM switches can create 300 VCs/s [Tec98]. However most switches have much lower signalling processor capacities. The earlier analysis shows that if the direct VC can be created, the Cut-Through approaches are preferred. However in regions where the switch signalling processor does not have sufficient capacity it is better to use the Hop-by-Hop approach or to minimise signalling network requirements by keeping direct VCs open for long periods. The signalling processor requirements of the WWW application when using the Cut-Through approach are considered further in Section 5.3.5.

5.3.4 Effect of the Number of Hops and Distance

The analysis presented to date has shown that the response time performance of approaches follow the same trends when servers are located in the U.S.A. or France. However, when the server is located in Australia, the performance of the approaches differs. As discussed in Section 5.3.1, the only difference between the Australian scenario and the other two scenarios is the ratio of the total distance travelled to the number of hops.

Table 5.8 shows the number of hops, total distance and the average number of kilometres per hop for each of the network scenarios analysed. This indicates
Table 5.8: WWW Scenario Hop and Distance Statistics

<table>
<thead>
<tr>
<th>Location</th>
<th>Number of Hops</th>
<th>Distance (km)</th>
<th>Km per Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Australia</td>
<td>9</td>
<td>800</td>
<td>89</td>
</tr>
<tr>
<td>U.S.A.</td>
<td>14</td>
<td>13000</td>
<td>929</td>
</tr>
<tr>
<td>France</td>
<td>18</td>
<td>23000</td>
<td>1278</td>
</tr>
</tbody>
</table>

that there is an order of magnitude difference between the km/hop ratio of the Australian scenario and the other scenarios. The effect of the km/hop ratio on the performance of the unicast delivery approaches is considered further in this section.

Our analysis showed that varying the km/hop ratio, has the same effect on approach response time performance, regardless of whether the total distance or the number of hops varies. Figure 5.9 examines the effect of the km/hop ratio by varying the distance per hop. Figure 5.9 shows that the response time performance of approaches is sensitive to the km/hop ratio. In particular, as the km/hop ratio increases, the difference between the response time performance of the approaches decreases. This occurs because as the km/hop ratio increases, the propagation delay, which affects all approaches equally, becomes increasingly dominant. Hence, the response time performance penalty of choosing the wrong approach is greater when the distance per hop is small. Furthermore Figure 5.9 shows that as the network utilisation increases the impact of the km/hop ratio lessens. This is because the propagation delay becomes a less dominant component of overall response time. That is, increasing the network utilisation increases the per hop delay but has no effect on the propagation delay.

To summarise this analysis has shown that the value of the km/hop ratio can have a significant impact on unicast delivery approach response time performance. In particular, the penalty of choosing the wrong approach is greatest when the distance per hop is small.

5.3.5 Cut-Through Approach Resource Requirements

The performance analysis of the unicast delivery approaches has shown that the Cut-Through approach provides significant response time performance gains over the Hop-by-Hop approach. This is particularly true if the direct VC already exists. This leads to the question of how often a direct VC must be created. This section addresses this issue and provides network providers with recommendations for tuning their network to best apply the Cut-Through unicast delivery approach.

In the Hop-by-Hop delivery approach, all packets travelling between two routers
Figure 5.9: Mean Response Time as the Distance Per Hop Varies Assuming 8 Hops
can be carried on the same VC, even if their original source and destinations differ. This means that the VC requirements and signalling resource requirements are low. Furthermore, the probability of an HTTP request having to wait for a hop-by-hop VC to be created is low, as one VC can be used by multiple source-destination pairs. Furthermore, new hop-by-hop VCs will only be created between a small number of intermediate routers (or a host and router), since many of the VCs on the hop-by-hop path between the source and destination will already exist. Therefore the delay and signalling resources associated with creating VCs for the Hop-by-Hop approach will be small.

In contrast, the Cut-Through approach requires a separate VC to be created for each source-destination pair. In the WWW network scenario considered in this chapter, this means that a separate direct VC is needed for each unique WWW server contacted by the proxy. In addition, if a given WWW server has not been contacted for some time, the direct VC may have timed out. Hence a new VC must be created when another HTTP request for that server arrives. Therefore the VC time-out period, or VC holding time, directly affects: (a) the probability that an HTTP request must wait for a VC to be created, (b) the frequency of cut-through VC creations, and (c) the number of active VCs. In this section these three factors are considered for a range of VC holding times, using the WWW access log from the University of Wollongong proxy to provide HTTP request inter-arrival information. This proxy serves 27 active users on average and transmits on average 1522 HTTP requests per hour to WWW servers. Traffic traces are divided into ten 24 hour time periods, and the analysis is performed on each trace. Results are shown with 95% confidence intervals and a 5% level of significance.

Figure 5.10 shows the performance of the Cut-Through approach in terms of the three factors described above as the holding time varies from one second to two hours. These results indicate that as the VC holding time increases beyond two minutes, the rate of decrease in the mean probability that a VC must be created drops. That is, once the holding time increases beyond two minutes the benefit of further increasing the VC holding time in terms of the number of requests waiting for VCs to be created reduces. To illustrate this further, the percentage of HTTP requests that must wait for a direct VC to be created are 23%, 11% and 10% for VC holding times of one, five, and ten minutes respectively.

As the holding time increases, the probability that a request must wait for a VC to be created reduces. This is because (1) users tend to visit several pages located on the same server in quick succession; and (2) each WWW page generates multiple HTTP requests, one for each embedded object comprising the page. However the
5. Unicast Delivery

(a) Probability of a Request waiting for a VC to be created

(b) Inter-Arrival Time between VC Creation Requests

(c) Number of Active VCs

Figure 5.10: Cut-Through Approach Resource Requirements as the VC Holding Time Varies
90th percentile remains constant with every HTTP request requiring a VC to be created, regardless of the holding time.

These results represent an upper bound on the probability that an HTTP request must wait for a direct VC to be created. This is because the analysis assumes that the VC inactivity timer begins when the HTTP request arrives at the proxy. In practice, the inactivity timer will not activate until the entire HTTP response reaches the proxy. The inter-arrival distribution of HTTP requests in the traffic traces are directly related to the response times experienced by users on that network. This is because users will select a new WWW hyperlink when the previous response has been received and they have been able to read it. This means the response times presented earlier in the chapter can not be employed to trigger the inactivity timer, since the log files were obtained in a different network environment.

The number of HTTP requests that trigger the creation of a direct VC determines the frequency of VC creation requests. Figure 5.10(b) shows the mean inter-arrival time between VC creation requests as the VC holding time varies. These results give network administrators an indication of the signalling network resources required by the Cut-Through approach when used by moderately sized proxies (such as the University of Wollongong proxy). The results indicate that even with a low holding time of five seconds, on average only one VC creation request is generated every five seconds. When the VC holding time is increased to ten minutes this reduces even further to one creation request on average every thirty seconds. Given, current switches can process 100 VCs/s, HTTP traffic generated by 3000 moderately sized proxies can be supported by one switch. The 90th percentile results are relatively insensitive to the VC holding time, with the arrival time varying between a fraction of a second to 3 seconds over the range of holding times considered.

Increasing the holding time also greatly increases the number of simultaneously active VCs as shown in Figure 5.10(c). The number of active VCs required starts to increase rapidly when the VC holding times exceed ten minutes. At a holding time of ten minutes, on average 50 active VCs are open, with a 90th percentile of 80. Currently, many vendors use a VC holding time of twenty minutes. At this operating point on average, 10% of HTTP requests must wait for a VC, VC creation requests occur every 35 seconds and there are on average 80 active VCs.

The previous discussion has shown that a network provider must trade the frequency of creating VCs against the number of active VCs depending on the particular capabilities of their network. In Figure 5.11, the optimal VC holding time is indicated, assuming that the frequency of VC creation requests and the
Figure 5.11: Optimisation of Holding Time Considering Signalling and VC Resources
number of active VCs are equally important. Based on this assumption, the analysis indicates that a VC holding time of five minutes should be used. Reducing the VC holding time from the current 20 minutes to 5 minutes will reduce the average number of active VCs by a factor of 2.5, and only increase the frequency of VC creations by a factor of 1.3.

By selecting the relative cost of signalling and VC resources and using the analysis presented in Figure 5.10, a network administrator can determine the optimal VC holding time for their network configuration. In Figure 5.12 the sensitivity of the optimal VC holding time is considered as the ratio of the signalling network cost to the VC cost varies between 0.1 and 10.0. This analysis indicates that the relative weighting of these parameters has a significant effect on the optimal holding time which ranges from 30 seconds to 20 minutes over the parameter weightings considered. Network administrators can use this information to select the optimal holding time given the relative costs in their particular network.

To summarise, this section has examined the resource requirements of the Cut-Through approach when carrying WWW traffic. This analysis has shown that the Cut-Through resource requirements of a moderately sized proxy can easily
be supported by current ATM network equipment. Furthermore, this analysis has produced guidelines which allow network administrators to select the optimal holding time for the Cut-Through approach given the relative costs associated with creating (and removing VCs) and maintaining a large number of VCs. When these costs are equally important, the analysis indicated an optimal holding time of five minutes.

5.3.6 Transmitting Smaller Volumes of Data

To date this chapter has focused on the performance of unicast delivery mechanisms when carrying WWW traffic. This was motivated by the fact that the WWW is currently the dominant Internet application. However, the WWW typically requires a large volume of data to be transmitted between the server and proxy. As discussed earlier, the main contributor to the response time performance difference between the Hop-by-Hop and Cut-Through approaches is the time to reassemble IP datagrams at each hop. In addition to transmission rate, reassembly delay also depends on packet size. In this section the performance of the Cut-Through and Hop-by-Hop approaches are compared for connection oriented applications (i.e. those that employ TCP) involving smaller volumes of data.

Figure 5.13 compares the response time performance of approaches assuming the server is located in the U.S.A., and a signalling network utilisation of 0.4. Both routers and switches are assumed to be able to process 100,000 PDUs/s. Hence the additional delay in the Hop-by-Hop and Hybrid approaches in Figure 5.13 is due primarily to datagram reassembly at routers. The application modelled creates a TCP connection and then transmits one application PDU of varying size. This analysis indicates that even with ten byte application PDUs the Cut-Through approach produces significantly lower response times than the Hop-by-Hop approach if the direct VC exists. Furthermore, in moderate to heavily loaded data networks, both the Hybrid and Buffered approaches also produce significantly lower response times than the Hop-by-Hop approach. At a data network utilisation of 0.3, the response time performance of the Hybrid and Buffered approaches only significantly differs from the Hop-by-Hop approach for application PDU sizes of at least 5000 bytes. However, it is unlikely that networks will operate at such low link utilisations. Hence this analysis indicates that the Cut-Through approach produces significantly lower response times than the Hop-by-Hop approach, even when the direct VC must be created.

Figure 5.13 also indicates that the response times produced by the Hop-by-Hop, Buffered and Cut-Through approaches increase markedly when the application PDU is sufficiently large to generate an MTU sized IP datagram and a smaller
Figure 5.13: Mean Response Time as the TCP Traffic Volume Varies (USA Server)
5. Unicast Delivery

datagram (i.e. if the application PDU is larger than 9140 bytes). This is because the TCP slow-start mechanism limits the number of unacknowledged datagrams that can be transmitted over the network simultaneously. However, the response time for the Hybrid approach increases much more slowly in this region. This is because both the TCP connection creation (SYN) messages and the initial MTU sized datagram are transmitted hop-by-hop. The delay produced by the transfer of these messages is much greater than the time to transmit the remaining datagrams via the direct VC. Hence transmitting additional datagrams has little impact on the overall Hybrid approach response time. This also means that the difference between the Hybrid and the other Cut-Through approaches decreases as the application PDU size increases.

To summarise, this analysis has shown that for any connection-oriented application the Cut-Through approach provides significant response time savings regardless of the size of the application PDU. Moreover, this performance difference is significant, even if the direct VC must be created via either the Hybrid or Buffered approaches.

5.3.7 Transmitting Connection-less Traffic

Although many Internet applications use the connection oriented transport protocol TCP, others, such as the Domain Name System (DNS) use the connection-less transport protocol UDP. In this section the response time performance of the unicast delivery approaches are compared when transmitting application PDUs of varying size using UDP.

The key difference between UDP and TCP is that UDP does not control the volume of traffic present in the network. This means that applications using UDP can transmit all of the datagrams comprising the application PDU sequentially without waiting for acknowledgment messages.

Figure 5.14 compares the response time performance of approaches when carrying connection-less data for a range of data network utilisations. The signalling network utilisation is assumed to be 0.4. The application being modelled transmits one application PDU of varying size into the network. Again this PDU will be broken into datagrams as required depending on its total size. This analysis indicates firstly that the response times achieved by the unicast delivery approaches are much lower when UDP is employed (Figure 5.14) rather than TCP (Figure 5.13). This is because UDP datagrams can be sent without waiting for a TCP connection to be established or for acknowledgment messages.

Figure 5.14 also shows that irrespective of data network utilisation the Hybrid approach has no significant response time performance benefits over the Hop-by-
Figure 5.14: Mean Response Time as the UDP Traffic Volume Varies

(a) Data Network Utilisation of 0.3

(b) Data Network Utilisation of 0.7

(c) Data Network Utilisation of 0.9
5. Unicast Delivery

This is because, even with application PDUs of 100,000 bytes, most of the datagrams comprising the application PDU are transmitted via the hop-by-hop path. This does not occur when TCP is employed because it takes a similar amount of time (one round trip delay) to create the TCP connection and the direct VC. Hence, when using TCP, the Hybrid approach transmits most data via the direct VC. In contrast when UDP is employed most of the data will be transmitted into the network before the direct VC can be created.

This analysis also indicates that both the Cut-Through and Buffered approaches produce significantly lower response times than the Hop-by-Hop and Hybrid approaches when the network is moderately to heavily utilised. This is because the time to create the direct VC is small compared to the time to transmit the application PDU hop-by-hop. When the network is lightly utilised, the Buffered approach produces higher response times than the Hop-by-Hop and Hybrid approaches until the application PDU size reaches 6000 bytes. This is because when the PDU is smaller than 6000 bytes, it is faster to transmit the data hop-by-hop than to create the direct VC.

To summarise, when the network is moderately to heavily utilised, the Buffered approach also produces significantly lower response times than the Hop-by-Hop approach. Furthermore, when using UDP the Buffered approach should be used to create the direct VC rather than the Hybrid approach. Indeed, when carrying connection-less traffic the Hop-by-Hop approach should be used in preference to the Hybrid approach, because the same response time performance can be achieved without wasting network resources creating a direct VC.

5.4 Conclusions

In this chapter the Hop-by-Hop and Cut-Through approaches for delivering unicast Internet traffic over ATM networks have been compared in various operating scenarios. This chapter has addressed each of the short-comings listed in Chapter 3 in current research comparing alternative unicast delivery approaches. Our findings for each open issue are summarised below.

- Determine whether the Cut-Through unicast delivery approach produces significantly lower delays than the Hop-by-Hop approach, particularly when the direct VC must be created.

This analysis has clearly indicated that the Cut-Through approach provides significant response time savings over the Hop-by-Hop approach, even if the direct VC must be created as long as the signalling network can support the
expected volume of signalling traffic. This finding holds regardless of the
volume of traffic transmitted and whether the application employs TCP or
UDP. Through the analysis presented in this chapter, it is also clear that
the greatest difference between the performance of approaches occurs when
the distance per hop is small.

- Determine in which network environments (if any) the VC and signalling
resource requirements of the Cut-Through approach are such that they can
not be supported by the network.

As stated above, the analysis indicated that the Cut-Through approach out-
performs the Hop-by-Hop approach if the signalling network can support
the expected volume of signalling traffic. Section 5.3.5 showed that the VC
and signalling resource requirements of the Cut-Through approach can easily
be supported by current ATM equipment for the workload of a moderately
sized proxy (such as the University of Wollongong proxy). Given, current
switches can process around 100 VC setups/s, HTTP traffic generated by
3000 moderately sized proxies can be supported by a single ATM switch.
New backbone ATM switches are being releases that can create on the order
of 300 new VCs/s [Tec98]. Hence, in all but the largest backbone ATM
networks the resource requirements of the Cut-Through approach can be
supported. However, it is also important to note that even though the
resource requirements of the Cut-Through approach can be met, the Hop-
by-Hop approaches require significantly less resources.

- Investigate whether the Hybrid approach provides better response time per-
formance than the Buffered or Retransmission approaches.

In contrast to the hypothesis commonly held within the literature, the analy-
ysis indicates that the Buffered approach should be employed to create direct
VCs rather than the Hybrid approach. This is because the Buffered ap-
proach is more robust than the Hybrid approach in terms of (1) the type
and volume of application traffic being transmitted, (2) the data network
utilisation, and (3) the available link bit rate. In circumstances where the
signalling network is unable to support the volume of signalling traffic it is
receiving the Hop-by-Hop approach should be used. In these situations the
Buffered approach produces extremely high response times and the Hybrid
approach wastes signalling resources by creating direct VCs which are never
used because all traffic is transmitted hop-by-hop. This chapter has also
shown that, due to the low volume of traffic that must be stored in the
Buffered approach for client-server applications, there is no benefit to be gained from employing the Retransmission approach.

- Analyse the sensitivity of unicast IP over ATM delivery approaches to the VC holding time.

Selection of the optimal holding time is particularly critical for the Cut-Through approach since it creates a separate VC for each source-destination pair. This investigation has produced a number of recommendations which allow network administrators to select the optimal VC holding time for supporting a moderately sized WWW proxy using the Cut-Through approach. The holding time can be selected in terms of the relative costs of maintaining a VC and creating a new VC. Furthermore, when the VC and signalling resources have equal importance the analysis found that a holding time of 5 minutes should be used rather than the currently popular holding time of 20 minutes. The analysis also showed that the optimal holding time is sensitive to the relative importance of signalling resources and the number of active VCs. If the weighting varies between 0.1 and 10.0, the optimal holding time varies between 30 seconds and 20 minutes.

In the Hop-by-Hop delivery approach, all packets travelling between two routers can be carried on the same VC, even if their original source and destinations differ. This means that the VC requirements and signalling resource requirements are low. These two factors mean the selection of the optimal holding time is less critical for the Hop-by-Hop approach. In general it is best to keep a VC open between two neighbouring routers semi-permanently, since in most network topologies there will be a continual flow of traffic between neighbouring nodes. In fact, if the Hop-by-Hop approach is employed it can be argued that PVCs should be employed rather than SVCs.

- Determine whether there is a significant delay associated with creating direct VCs via the Cut-Through approach and from this whether the Label Switching approach significantly reduces the direct VC creation time.

The response time analysis has shown that in some circumstances there can be a significant performance difference between the Buffered and Cut-Through approaches, particularly when the link utilisation is low, or the volume of data traffic that must be transmitted is low (e.g. for UDP based unicast traffic). Hence this analysis indicates that the Label Switching approach can produce significantly lower response times than the Cut-Through
approach in some network environments. However in the majority of sce­
narios considered there was not a significant difference between the Cut-
Through and Buffered approach response time performance. Furthermore,
the resource analysis showed that current ATM equipment can support the
resources required to create VCs via traditional ATM protocols. Further
analysis of specific Label Switching proposals is required to confirm whether
they have significant benefits over the Cut-Through approach. However, our
analysis has shown that in many operating environments, there is no signif­
icant response time increase caused by creating a direct VC, and secondly,
that the signalling resource requirements of the Cut-Through approach can
be supported by existing ATM switching equipment.
Chapter 6

Multicast Delivery

6.1 Introduction

As discussed in Chapter 3, three key issues must be considered when investigating the relative merits of multicast delivery approaches: (1) delay performance; (2) VC requirements; and (3) dynamic multicast group support. This chapter examines the delay performance of alternative multicast delivery approaches via the methodology described in Chapter 4. Each of the multicast delivery delay issues highlighted in Section 3.6.2 are addressed.

To create an Internet multicast delivery system over ATM networks, intra-subnet, inter-subnet and multicast forwarding tree approaches are required. Sections 3.3, 3.4, and 3.5 described alternative approaches in each of these categories. Table 3.2 lists all 22 possible multicast delivery systems.

Chapter 3 highlighted the main shortcoming in the area of multicast delivery over ATM is a lack of quantitative analysis. This chapter analyses the performance of the entire multicast delivery system when alternative approaches from each of the categories above are combined. Approaches are compared in terms of their response time performance in this chapter. The approach VC requirements and support of dynamic multicast groups are considered in Chapters 7 and 8 respectively.

As discussed in Chapter 2 the future Internet will support both best-effort and real-time applications. Hence the performance of the multicast delivery system must be considered when supporting a variety of traffic types. Distributed Interactive Simulation (DIS) combines a number of traffic types: short update messages with strict delay requirements; voice traffic; and large file transfers with best-effort delivery requirements. Hence comparing multicast delivery approaches in the presence of DIS traffic extends directly to many other multicast applications. For
example, voice conferencing, CSCW and multi-player games. Therefore in the remainder of this chapter, DIS is employed to compare alternative approaches for providing Internet multicast delivery over ATM.

There is an increasing demand for real-time Internet services, e.g. voice or video conferencing. Integrated Services and Differentiated Services have been developed to enable: (1) resource requirements to be communicated to network nodes, and (2) to enable these network nodes to process each traffic flow in a manner that suits its resource requirements. In this chapter we investigate whether it is possible to meet the real-time performance requirements of message flows, if all traffic is carried in a best effort fashion. That is, if the resource requirements of delay sensitive traffic flows can be met when they must share resources with best effort, delay insensitive traffic. This will allow us to determine in which situations real-time traffic has to be treated separately, and hence protocols such as RSVP and per traffic type scheduling is required. The analysis of delay sensitive traffic flows is presented in terms of percentiles. This is because delay percentiles give greater information about the number of packets that are likely to arrive within the delay constraints of real-time applications (e.g. a voice or video conference), and hence the need to reserve resources for real-time traffic.

Section 6.2 provides an overview of DIS, followed by a description of how the DIS application is modelled. The network model and performance measures employed in this chapter are also detailed. Section 6.3 compares the response time performance of multicast delivery approaches as a variety of network parameters change. The findings and recommendations of this chapter are then summarised in Section 6.4.

6.2 Analysis Methodology

6.2.1 DIS Overview

Distributed Interactive Simulation (DIS) refers to a group of applications that simulate the behaviour of interacting objects in a virtual environment (e.g. a battlefield) [MilTho95]. DIS is recognised by many researchers as placing the highest demands on multicast delivery mechanisms of all currently available multicast applications. This is because DIS supports the exchange of information between potentially tens of thousands of geographically dispersed users. These hosts can be distributed over many multicast groups, and multicast group membership may change frequently.

DIS also involves a wide variety of media including video, audio, small update messages and large files. Many of these traffic flows have strict delay requirements.
For instance, the DIS standards [IE1278.1],[IE1278.2] state that the maximum acceptable latency for update messages for tightly coupled interactions (e.g. high performance aircraft in a dog fight) is 100 ms. For less tightly coupled interactions (e.g. voice radio communication) 300 ms end-end delay is acceptable.

The choice of DIS for comparing multicast delivery approaches has the additional advantage that many of its characteristics are similar to other applications. For example, DIS often involves the transfer of voice between multiple users, similar to audio conferencing. Similarly, DIS involves large file transfers, (e.g. weather updates). Hence the performance of this traffic flow is similar to multicast file transfer, CSCW or distributed file systems support. Thus the choice of DIS enables us to determine: (1) how well each multicast delivery approach supports extremely demanding multicast applications, and (2) how well each approach supports real time multicast flows that are transported in a best effort fashion. The next section describes the DIS application scenario assumed in the remainder of the chapter.

6.2.2 DIS Traffic Model

[SSM96] describes several DIS scenarios, ranging from a LAN based DIS, to a global multi-player game modelling a large historical battle. However, only the LAN based DIS scenario is described in detail. This chapter extends the scenario by distributing the hosts across a European wide area ATM network rather than a LAN. Future DIS is more likely to occur over WANs because it avoids the need for participants to move to one location. This reduces travel costs and allows DIS to occur when timeliness or political issues do not allow participants to move to one site [PulWoo95].

The DIS scenario applied in [SSM96] involved simulation entities including ground vehicles, rotary wing aircraft, high speed aircraft and one airborne warning and control (AWAC) aircraft. Table 6.1 describes the 171 simulation entities present in this DIS scenario and how they are distributed across ten physical computers (hosts).

<table>
<thead>
<tr>
<th>Entity</th>
<th>Number</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Vehicle</td>
<td>160</td>
<td>Hosts 1,2,3,4</td>
</tr>
<tr>
<td>Rotary Aircraft</td>
<td>6</td>
<td>Host 5</td>
</tr>
<tr>
<td>High Speed Aircraft</td>
<td>3</td>
<td>Hosts 6,7,8</td>
</tr>
<tr>
<td>Airborne Control Aircraft</td>
<td>1</td>
<td>Host 9</td>
</tr>
<tr>
<td>Weather Station</td>
<td>1</td>
<td>Host 10</td>
</tr>
</tbody>
</table>

Table 6.1: Number of DIS Entities and Their Location
Each of these entities generate several message flows which can be categorised in terms of their size, arrival characteristics and delay requirements. The following four multicast message categories are assumed in the remainder of this chapter:

**Type I**
- Small messages
- Poisson arrivals
- Strict 100 ms end-end delay requirement.

**Type II**
- Small messages
- Bursty arrivals
- Strict 100 ms end-end delay requirement.

**Type III**
- Small messages
- Bursty arrivals
- Strict 300 ms end-end delay requirement.

**Type IV**
- Large messages
- Deterministic arrivals
- Delay insensitive.

Note in the analysis performed in this chapter, it was found that there was no significant difference in the performance of approaches when either type I, II, or III messages are employed. However there are significant differences in approach performance when type IV messages are employed. Hence in the remainder of this chapter results are only presented for type I and type IV message flows. Examples of multicast applications that would employ these types of flows are file transfers for CSCW, forwarding news, sports results etc (e.g. push style services), or multiplayer games. It is important to note that even though the chapter only contains results for type I and IV messages, the traffic flowing over the analysed network is a mix of all four traffic types as detailed in Table 6.2.

Table 6.2 shows the statistics assumed for each DIS message flow. All DIS messages are transmitted using UDP since this is the most widely used multicast transport protocol. TCP in its current form is only applicable to unicast message flows. The development of reliable multicast transport protocols is a study area in its own right [Mul98] and not considered further here. Hence in this thesis only UDP is employed to transport multicast traffic. Unless indicated otherwise the figures in Table 6.2 were determined from the DIS LAN scenario description in [SSM96].
6. Multicast Delivery

<table>
<thead>
<tr>
<th>Message Flow</th>
<th>Distribution</th>
<th>Size (bytes)</th>
<th>$\lambda$ (msgs/s)</th>
<th>Arrival S.C.V.</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ground Vehicle Update</td>
<td>All entities</td>
<td>144</td>
<td>2.0</td>
<td>1.0</td>
<td>I</td>
</tr>
<tr>
<td>Rotary Aircraft Update</td>
<td>All entities</td>
<td>144</td>
<td>6.0</td>
<td>1.0</td>
<td>I</td>
</tr>
<tr>
<td>Rotary Aircraft Fire</td>
<td>All entities</td>
<td>88</td>
<td>0.4</td>
<td>236a</td>
<td>II</td>
</tr>
<tr>
<td>Rotary Aircraft Detonate</td>
<td>All entities</td>
<td>104</td>
<td>0.4</td>
<td>236a</td>
<td>II</td>
</tr>
<tr>
<td>High Speed Air Updates</td>
<td>All entities</td>
<td>144</td>
<td>7.5</td>
<td>1.0</td>
<td>I</td>
</tr>
<tr>
<td>Background Voice</td>
<td>All entities</td>
<td>64</td>
<td>22.0</td>
<td>18.1</td>
<td>III</td>
</tr>
<tr>
<td>Aircraft Radio</td>
<td>Aircraft only</td>
<td>64</td>
<td>13.3</td>
<td>26.6</td>
<td>III</td>
</tr>
<tr>
<td>Weather Update</td>
<td>All entities</td>
<td>$6.25 \times 10^6$</td>
<td>0.0017</td>
<td>0.0</td>
<td>IV</td>
</tr>
</tbody>
</table>

"We assume for the fire and detonate messages that a burst occurs on average every 300 s and lasts 2 s with 60 packets/s generated in the active state.

Voice statistics from [SW86]

The silent period is assumed to be double the silent period of typical voice conversations.

Table 6.2: DIS Message Flow Statistics

Given an ATM MTU of 9180 bytes (including UDP and IP headers) the DIS weather update message will be broken into 68 MTU sized datagrams and one small datagram as shown in Table 6.3. Hence the arrival rate of small and MTU sized weather datagrams will be the same and 68 times greater than the overall weather update message arrival rate, respectively.

The variation in the arrival process of small weather datagrams is the same as for the overall weather message arrival process. To capture the burstiness of the arrival process for MTU sized datagrams, this message flow is modelled as a two state process with a silent period of 10 minutes (the inter-arrival time between weather messages). The s.c.v. of the arrival process is calculated from Equation 6.1 (Equation 2 from [SW86]) where: $\rho$ is the probability that the packet is not the last in the active period (67/68 in this case); $\tau$ is the inter-arrival time between packets in the active period (0 in this case since all datagrams comprising the weather message are generated simultaneously); and $\beta$ is the silent period duration (600s). This gives an arrival s.c.v. $c_a^2 = 135$ as shown in Table 6.3.

$$c_a^2 = \frac{1 - \rho^2}{[\tau \beta + (1 - \rho)]^2} \quad (6.1)$$

6.2.3 Network Model

The Joint ATM Experiment on European Services (JAMES) network [Con96] is used for the majority of the analysis in this chapter. This network was selected because: (1) the network topology is known [Con96], and (2) it is a good example of a typical future ATM WAN where applications such as DIS may be employed. The
### Table 6.3: Weather Message Datagram Flows

<table>
<thead>
<tr>
<th>Weather Datagram Flow</th>
<th>Number per Weather Message</th>
<th>Size (bytes)</th>
<th>λ (datagrams/s)</th>
<th>Arrival S.C.V.</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU Sized Datagram</td>
<td>68</td>
<td>9152</td>
<td>0.1130</td>
<td>135</td>
</tr>
<tr>
<td>Small Datagram</td>
<td>1</td>
<td>2664</td>
<td>0.0017</td>
<td>0</td>
</tr>
</tbody>
</table>

![Figure 6.1: JAMES Network Topology as of September 1996 [Con96]](image-url)
JAMES network topology, reproduced from [Con96], is shown in Figure 6.1. The distances between network nodes were estimated using the geographic distance server [Onl]. To examine the effect of the number of hops and distance between senders and receivers the network topology is varied in this chapter. Hence the comparison of multicast delivery approaches is applicable to a variety of networks not just the specific JAMES network topology.

As described in Section 3.4, several multicast delivery approaches break the network into multiple subnets. This chapter assumes there is a subnet associated with each JAMES network node consisting of one switch, up to ten hosts, a local MARS and MCS as shown in Figure 6.2. Thus the JAMES network model can be used to examine the performance of an entire multicast delivery system. Each subnet ATM switch is assumed to be attached to the JAMES network node via a 100 m link. Hosts, MARS and MCSs are assumed to be attached to their local LAN switch via 50 m links.

The default transmission and processing delays assumed in this analysis are shown in Table 6.4. One can see that all data traffic is carried over one UBR VC\(^1\), hence the suitability of best effort delivery for traffic with real-time requirements can be considered. Throughout this chapter, many of these parameters are varied to investigate their effect on the performance of alternative multicast delivery approaches. However, from the default values one can see that the router and MCS processing delays are assumed to be comparable to ATM switch delay, with the only difference being that routers and MCSs must reassemble entire packets rather than just single cells. Hence any difference between approaches is caused by a difference in delivery paradigm rather than processing delay differences which

\(^1\text{UBR was selected since it is the most commonly used ATM service class for carrying Internet traffic}\)
will change as hardware capabilities improve.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Propagation Delay</td>
<td>5 ( \mu s/\text{km} )</td>
</tr>
<tr>
<td>Data Link Rate</td>
<td>UBR 5 ( M\text{bits/s} )</td>
</tr>
<tr>
<td>Signalling Link Rate</td>
<td>CBR 1 ( M\text{bits/s} )</td>
</tr>
<tr>
<td>Switch Processing Delay</td>
<td>10 ( \mu s )</td>
</tr>
<tr>
<td>Router Processing Delay</td>
<td>10 ( \mu s )</td>
</tr>
<tr>
<td>MCS and RP Processing Delay</td>
<td>10 ( \mu s )</td>
</tr>
<tr>
<td>Signalling Processor Speed</td>
<td>1000 ( V\text{Cs/s} )</td>
</tr>
</tbody>
</table>

Table 6.4: Multicast Transmission and Processing Delay Assumptions

Background traffic is assumed to be present on both data and signalling networks. Signalling network background traffic is modelled by VC creation messages (setup, connect and connect acknowledgment). In the current Internet environment, WWW traffic is the dominant traffic type. Hence this chapter assumes WWW background traffic for data networks. The traffic characteristics of the WWW background traffic (comprising TCP connection, HTTP request and HTTP response messages) are taken from the statistics presented in Section 5.2.2. This chapter compares alternative multicast delivery approaches via the performance analysis methodology detailed in Chapter 4.4. Note subnet MCSs and MARS are modelled in the same way as hosts (see Section 4.5).

Table 6.1 described a DIS comprising ten hosts. The hosts are randomly distributed across the subnets comprising the JAMES network. However, to allow consistent comparison of approaches as network parameters varies, the location of hosts are fixed throughout the analysis to the randomly selected locations shown in Figure 6.3. Furthermore all traffic is assumed to follow the shortest path between each pair of hosts.

In approaches that employ the Shared Forwarding Tree approach (where all senders transmit multicast traffic to one network location which then forwards the data to all receivers), this core node or rendezvous point (RP) is assumed to be located at Köln. Köln was selected because it is the wire-line centre of the JAMES network.

Performance Measures

The focus of this chapter is the delay performance of alternative approaches for delivering multicast application traffic. As outlined in Chapter 3 the major delay issues that must be examined are to:

- Determine in which situations (if any) there is a significant delay difference
Figure 6.3: DIS Host Locations on the James Network
between the VC Mesh and MCS intra-subnet multicast delivery approaches.

- Examine the sensitivity of the MCS approach, in terms of delay, to the location of the MCS, and the Shared Forwarding Tree approach to the location of the Rendezvous Point.

- Investigate whether the Hop-by-Hop inter-subnet delivery approach produces significantly higher delays than the alternative direct VC based inter-subnet approaches.

- Investigate whether the Boundary variants of the Single Subnet, LSR and NHRP Router approaches have significantly higher delays than the Cut-Through variants of these approaches.

- Quantify the delay performance differences between Shared or Source multicast Forwarding Trees when employed in an ATM environment.

- Determine which of the multicast delivery component (intra-subnet, inter-subnet or forwarding tree approach) choices has the greatest impact on the delay performance of the multicast delivery systems.

To address these issues effectively, representative network topologies and multicast applications must be considered. Distributed Interactive Simulation (DIS) is an extremely demanding multicast application with stringent delay requirements. As such, DIS is a good example application to determine the relative delay performance of alternative multicast delivery approaches. In particular the stringent delay performance requirements can be used as a measure to determine when there is a significant difference in the delay performance of multicast delivery approaches.

For real-time applications it is not critical when most of the packets arrive. The parameter that is critical is whether packets arrive within the delay bound. This is because any packet arriving outside the delay bound can not be used by the application, and hence can impair the perceived performance of the application. Given type I, II, and III messages have strict end-end delay requirements, the response time performance of approaches when carrying these messages is presented via the response time 95th percentile rather than mean delay. The exact percentage of packets an application can lose (due to packet loss or packets arriving too late), before the perceived quality of the application is impaired depends on the error resilience of the application (e.g. the type of video or audio codec) employed. However many video and audio applications can adapt to several percent of loss, hence the choice of 95th percentiles. The percentiles are determined from
the response time distribution approximation process described in Section 4.4.4. The objective of this analysis is to determine if the real-time delay requirements of these message flows can be met by any of the multicast delivery approaches, and if so in which network scenarios.

Type IV messages have no strict delay constraints, thus mean response times are presented to compare approaches when carrying type IV messages. These message flows are included in the mix of traffic analysed to determine (a) how they impact the performance of the delay constrained message flows, if all of this traffic is carried in a best effort manner and (b) to investigate the relative delay performance of approaches when large messages are involved. Although bulk transfer applications do not have strict delay requirements the end user is often interested in receiving the information as quickly as possible. Hence the relative performance of approaches for bulk transfer message flows may also impact the decision of the optimal multicast delivery approach.

For all types of message flows the variation in the time traffic arrives at each receiver is very important, particularly for interactive multicast applications, e.g. voice or video conferencing, DIS, or multi-player games. This is because high jitter causes synchronisation to be lost between the communicating hosts. This makes it difficult to maintain conversations, or up to date DIS or game state information. Hence the jitter in the time traffic arrives at different receivers is also considered in this chapter. Results are only presented for one message flow from each of the message categories described above, results for other flows within each message category were similar.

To summarise, the remainder of this chapter compares the delay performance of the alternative multicast delivery approaches. A DIS application, involving message types I, II, III and IV is employed to provide the multicast traffic mix. Each of the delay issues listed above are addressed, via end-end mean delay, end-end delay 95th percentiles and the jitter in the time traffic arrives at different receivers.

6.2.4 Multicast Delivery Approaches

As described in Chapter 3, the following inter-subnet delivery approaches have been proposed:

- Hop-by-Hop
- Cut-Through - Single Subnet
- Cut-Through - LSR
6. Multicast Delivery

- Cut-Through - NHRP Router
- Boundary - Single Subnet
- Boundary - LSR
- Boundary - NHRP Router

The Hop-by-Hop approach reassembles traffic at every subnet boundary. In contrast the Cut-Through Single Subnet, LSR, and NHRP Router approaches all create direct VCs across the ATM network (Note, as discussed in Chapter 3 the LSR approach does not employ end-end signalling, instead each portion of the VC between neighbouring LSRs is created separately. However the overall effect is to create an end-end direct VC). These three approaches differ only in how they create the direct VCs. Hence these three approaches will exhibit the same response time performance once the necessary VCs are created. Therefore this chapter refers to this set of approaches as the Cut-Through approach in the response time analysis.

Furthermore, Chapter 3 described the Boundary variants of the Single Subnet, LSR and NHRP router approaches where direct VCs are created across the core network, but traffic is reassembled at the boundary between the core network and the local area networks. The Boundary approach is a compromise between the Cut-Through and Hop-by-Hop approaches. Again in terms of response time performance there is no difference between the Boundary Single Subnet, LSR and NHRP approaches. Therefore they are referred to collectively as the Boundary approach in the remainder of this Chapter. Thus the approaches listed in Table 3.2 reduce to the following ten multicast delivery approaches in terms of response time performance.

- Cut-Through, Source Forwarding Tree
- Cut-Through, Shared Forwarding Tree
- Hop-by-Hop, Source Forwarding Tree, VC Mesh
- Hop-by-Hop, Source Forwarding Tree, MCS
- Hop-by-Hop, Shared Forwarding Tree, VC Mesh
- Hop-by-Hop, Shared Forwarding Tree, MCS
- Boundary, Source Forwarding Tree, VC Mesh
- Boundary, Source Forwarding Tree, MCS
6. Multicast Delivery

- Boundary, Shared Forwarding Tree, VC Mesh
- Boundary, Shared Forwarding Tree, MCS

6.3 Response Time Performance Analysis

Chapter 3 hypothesised that multicast delivery systems employing Source Forwarding Trees, the VC Mesh intra-subnet approach, or that minimise the need to reassemble datagrams at routers should produce significantly lower response times than alternative multicast delivery approaches. The purpose of this section is to investigate (1) whether alternative multicast delivery systems are able to meet real-time application delay constraints when all traffic is carried in a best effort fashion and (2) determine whether there is a significant response time performance difference between approaches, and if so in which circumstances. This will be achieved by examining the sensitivity of approaches to the following network parameters:

- Link Utilisation
- Available Link Bit Rate
- Background Traffic Arrival S.C.V.
- Background Traffic Message Size
- Location of the Rendezvous Point
- Number of Senders
- Number of Receivers

In multicast delivery systems, the response times observed by different receivers will vary. In this section, the response time for the receiver that takes the longest to receive traffic from the sender (the "furthest" receiver) is presented. This represents the time when the message has arrived at all receivers. The jitter in the response times observed by different receivers (i.e. the time difference between a message arriving at the closest receiver and furthest receiver) is also examined.

6.3.1 Effect of the Data Network Utilisation

This section tests the hypothesis detailed above by comparing the response time performance of multicast delivery approaches as the data network link utilisation varies. This is achieved by varying the volume of background traffic present in the
JAMES network. One expects the response times achieved by all approaches to increase with increasing data network utilisation. The objective of this section, in addition to answering the two questions raised above, is also to observe whether some approaches are more sensitive to data network utilisation than others.

Figures 6.4 and 6.5 compare the mean and 95th percentile response times achieved by each multicast delivery approach when carrying type I messages as data link utilisation varies. Similar results were obtained for type II and type III messages. Both figures clearly show that no approach meets the type I message 100 ms delay constraint when the network is heavily utilised. Indeed most approaches 95th percentile results exceed this delay constraint when the link utilisation is between 0.75 and 0.825. Given this is a common operating point for many networks carrying IP traffic [Ste96], one can argue that the performance difference between approaches that exceed the 100 ms delay constraint earlier than others is therefore significant. The mean response time performance of results shows that the on average approaches do not exceed the delay constraint until the network is more heavily utilised (between 0.85 and 1.0). That is, compared to the 95th percentile results, the relative performance of approaches is the same, just shifted to a higher utilisation. Given, the number of packets that arrive within the delay bound has a greater impact on the perceived quality of delay sensitive applications than the average response time, the remainder of the analysis focuses on delay percentile results only.

The Hop-by-Hop, MCS, Source and Shared Forwarding Tree approaches provide the highest response times, with 95th percentiles exceeding the delay constraint at a link utilisation of 0.75. These are followed by the Boundary MCS Source and Shared Forwarding Tree approaches which exceed the delay constraint around 0.78 link utilisation. The Boundary VC Mesh and Cut-Through approaches do not exceed the delay requirements until the link utilisation reaches 0.825. Thus this initial analysis indicates that there is only a significant difference between the delay performance of the Hop-by-Hop inter-subnet delivery approach compared to the Cut-Through approaches when the MCS intra-subnet approach is employed and the network is highly utilised. Moreover there is little difference between the Hop-by-Hop and Boundary approaches when employed in conjunction with the MCS or VC Mesh intra-subnet approaches until the link utilisation exceeds 0.8 (even then the difference is on the order of a few tens of milliseconds). This seems to contradict the general belief in the literature that the Hop-by-Hop approach produces significantly higher delays than the direct VC based inter-subnet approaches.

The four approaches that employ the intra-subnet MCS approach overload ear-
Figure 6.4: Type I Message Mean Response Time as Data Network Utilisation Varies
Figure 6.5: Type I Message Response Time 95th Percentile as Data Network Utilisation Varies
6. Multicast Delivery

lier (at a link utilisation of 0.95), and produce higher delays than the remaining six approaches. This raises the issue of why the MCS intra-subnet delivery approach significantly increases response time performance compared to approaches employing VC Mesh intra subnet delivery and those that do not break the network into subnets. In particular it is interesting that there is a significant difference between the MCS and VC Mesh intra-subnet approaches yet Figure 6.5 shows that there is no significant difference between employing Shared and Source Forwarding Trees. This is even though the MCS and Shared Forwarding Tree approaches follow the same design philosophy, as do the VC Mesh and Source Forwarding Tree approaches.

We hypothesise that the impact of forcing multicast traffic through one point, whether a Shared Forwarding Tree RP, or an MCS, is directly related to the number of interfaces to that node. For instance, as can be seen from Figure 6.1, Köln (where the Shared Forwarding Tree RP is located) has eleven interfaces, ten from other core network nodes and one from its locally attached subnet. In contrast each subnet MCS has only one interface. Hence the performance penalty of delivering traffic via the subnet MCS is greater than delivering the traffic via the RP. This hypothesis is investigated further in Section 6.3.4.

Figure 6.5 also shows that there is little difference between the Boundary VC Mesh and Cut-Through approaches. In terms of topology the only difference between these approaches is that the Boundary VC Mesh multicast delivery system reassembles traffic on the boundaries between the core JAMES network and the source and destination subnets, whereas the Cut-Through approach does not. This therefore indicates that the use of switches or routers at the edge of the core network does not significantly affect response time performance. However, employing routers at every core network node, rather than just at the core network boundaries does appear to have more impact (this can be seen by comparing the Hop-by-Hop and Boundary approaches under high link utilisations, particularly when VC Mesh intra subnet delivery is employed). Moreover, in many senses this is the best case Hop-by-Hop analysis. The only difference between routers and switches in this analysis is the requirement to reassemble traffic at routers, with the processing delay assumed to be identical. However, in practice the processing delay of routers is often higher than in switches (e.g. processing delay to reassemble and segment packets and for routing table lookup).

Type I messages are only 144 bytes in length, corresponding to one IP datagram which is broken into three ATM cells. Hence the time to reassemble type I messages is small. We hypothesise that as the size of the IP datagrams being reassembled increases, the use of routers will have a greater impact on response
time performance. This will be considered further in Section 6.3.3.

In addition to the response time performance of approaches, another important factor is the difference, or jitter, in the delays seen by different receivers depending upon their relative location to the sender. This performance measure must also be minimised to keep the receivers synchronised. Figure 6.5 presents the maximum response time seen by all receivers of that message flow. Figure 6.6 compares the response time jitter of approaches, i.e. the difference between the response time observed by the receiver closest to the sender, compared to the receiver furthest from the sender.

Figure 6.6 clearly shows that the jitter produced by approaches is significant, particularly at high link utilisations where it is half of the magnitude of the total response times achieved by the alternative multicast delivery systems. One may expect that approaches employing either the MCS approach or Shared Forwarding Trees would produce lower response time jitter than their counterparts since they require all traffic to be forwarded via an MCS or RP. However, Figure 6.6 clearly shows that this is not the case. There are several reasons for this: Firstly, the choice of employing the VC Mesh or MCS intra-subnet approach will have little impact on jitter due to the distances and number of hops involved within a subnet compared to the JAMES network as a whole (the MCS is connected to the switch by a 50m link, and the subnet comprises only one switch). Secondly some hosts are closer to the RP than others. For instance, Luxembourg is only 152km and one hop from Köln, whereas Oslo is 1962km and 3 hops from Köln. Hence, delivering all traffic via the RP, does little to smooth the response times observed by different receivers.

Given the magnitude of the jitter shown in Figure 6.6, and its detrimental impact on group communication, those approaches which produce the lowest jitter should be favoured. Figure 6.6 shows that the approaches producing the lowest jitter are the Cut-Through and Boundary approaches (regardless of the Forwarding Tree approach), with jitter between 5 and 10ms until the link utilisation reaches 0.95.

We also considered the performance of multicast delivery systems when delivering delay insensitive traffic. In this case the objective is to minimise type IV message response times. Figure 6.7 shows that the time to delivery the 625 Kbyte Type IV message is between one and two seconds regardless of the approach. In general the response time performance of approaches follows similar trends as for type I messages (see Figure 6.5). However, one difference is that when delivering type IV messages, the Hop-by-Hop VC Mesh approaches produce higher response times than the Boundary MCS approaches when link utilisation is below 0.9. As
6. Multicast Delivery

Figure 6.6: Type I Message Response Time Jitter as Data Network Utilisation Varies
discussed above, type I messages are small. However the type IV message assumed in this analysis must be broken into 68 MTU sized (9180 byte) IP datagrams which are each broken into 192 ATM cells. Figure 6.7 shows that the delay penalty associated with deploying routers and hence reassembling large datagrams at each core network node, outweighs the delay associated with reassembling only at the network edge, even when the MCS intra-subnet approach is employed.

Figure 6.7 indicates that there is not a significant difference in performance between approaches when delivering large messages. Even when the link utilisation exceeds 0.9, all approaches take between 1.25 and 1.5 seconds to deliver the type IV message. This is because when large messages are involved, most of the delay is due to the time to transmit all of the IP datagrams comprising the application PDU into the network. This delay is common to all approaches. Thus as the number of IP datagrams comprising the application PDU increases, the significance in the difference between the delay performance of multicast delivery approaches begins to decrease.

Even though type IV messages are delay insensitive, jitter is still an important issue, since if one DIS participant gains update information significantly earlier than another player, they have an unfair advantage. In this context 'significantly earlier' refers to the situation where one DIS participant can make use of the information before other participants.

Figure 6.8 shows that until the link utilisation increases above 0.9, all Boundary and Cut-Through approaches produce response time jitter on the order of ten milliseconds. However the Hop-by-Hop approaches which employ a router at every JAMES network node produce a response time jitter on the order of 50ms, i.e. five times greater than the other approaches. Therefore, when transmitting large multicast datagrams, the Hop-by-Hop approaches should not be employed, if response time jitter is critical to the performance of the application. All other multicast delivery approaches have comparable performance.

This section has shown that in terms of data link utilisation, the Boundary VC Mesh and Cut-Through multicast delivery systems provide the lowest response times, regardless of whether Shared or Source Forwarding Trees are employed. The analysis has also shown that all multicast approaches are sensitive to whether the MCS or VC Mesh intra-subnet approach is employed. This leads to the hypothesis that the number of links connecting the MCS or RP to the remainder of the network has a significant impact on the delay performance of MCS and Shared Forwarding Tree approaches. This analysis has also highlighted that the Hop-by-Hop approaches only produce significantly higher delays than other approaches when the network is heavily utilised, particularly when transporting large mes-
Figure 6.7: Type IV Message Mean Response Time as Data Network Utilisation Varies
Figure 6.8: Type IV Message Response Time Jitter as Data Network Utilisation Varies
6. Multicast Delivery

sages. Indeed the choice of intra-subnet delivery approach had a greater impact on delay performance than the choice of inter-subnet delivery approach.

6.3.2 Effect of the Data Network Link Bit Rate

The previous section compared the performance of multicast delivery systems as the data link utilisation varied, assuming a link bit rate of 5 Mbits/s. However, it is not clear how sensitive the multicast delivery approaches are to the available link bit rate. The available link bit rate affects the time to transmit data from each network node. Therefore we expect the difference between approaches that employ routers and switches to decrease as the available link bit rate increases. This section examines this hypothesis, to determine in what regions there are significant differences between the approaches.

In many ways, increasing the link bit rate is similar to decreasing the volume of traffic being carried on the links. Hence a plot of response time against increasing link bit rate will have the opposite trend to the plots shown in Section 6.3.1. However, in practice, when the link bit rate increases, this is often because the total volume of traffic being supported is higher. Thus it is sensible to consider the response time performance of multicast delivery systems as link bit rate increases, but link utilisation remains static. Many networks operate at moderate to high link utilisations, particularly when carrying Internet traffic, hence this section assumes a link utilisation of 0.8.

Figure 6.9 compares the response time performance of approaches as the link bit rate varies between 1 and 50Mbits/s. At higher link bit rates, all approaches overload (indicated by the vertical line in the graphs) because the volume of traffic being transmitted on the network exceeds the processing capabilities of both switches and routers. Figure 6.9 also shows that there is a significant difference between systems employing the MCS intra-subnet delivery approach and those employing the VC Mesh or no intra-subnet approach. Indeed, the non-MCS approaches only exceed the 100ms type I message delay constraint at a link rate of 10Mbits/s. In contrast the Hop-by-Hop and Boundary MCS approaches only avoid exceeding the delay constraint when the link bit rate exceeds 30Mbits/s. For real-time applications with strict delay bounds this could have serious implications on the perceived performance of the application. Thus again this analysis indicates that the most crucial decision factor is the choice of intra-subnet delivery approach. This has a far greater impact on overall delay than the choice of inter-subnet or forwarding tree mechanism.

When available link bit rate is low, two factors come into play. Firstly, transmission delay becomes dominant and secondly, the foreground traffic (i.e. the DIS
Figure 6.9: Type I Message Response Time 95th Percentile as Data Link Bit Rate Varies
6. Multicast Delivery

message flows) has a greater influence on overall network performance. Figure 6.9 shows that as our hypothesis anticipated, when link bit rate is low, the difference between approaches is greater. This is because the delay penalties associated with (1) reassembling traffic at routers and (2) forwarding traffic via an MCS or RP rather than via the shortest path are higher in this environment. As link bit rate increases, the difference between approaches decreases. However, there appears to be a consistent 20 ms difference between approaches employing MCS intra-subnet delivery and other approaches when the link bit rate is greater than 10 Mbits/s. This is due to the volume of multicast traffic queuing at the subnet MCSs which is relatively unaffected by the link bit rate or the volume of background traffic (it is more dependent on the processor capabilities of the MCS).

Figure 6.9 also shows that the response times produced by multicast delivery systems employing the Cut-Through inter-subnet, or VC Mesh intra-subnet approaches increases as link bit rate increases to 10 Mbits/s, and then decreases when the link bit rate increases further. As the link bit rate increases, the volume of background traffic present on the network increases, increasing processor queuing delays, since the processing capabilities of the network elements (routers and switches) are not modified as the link bit rate increases. At the same time transmission delays decrease due to the higher link bit rate. When the link bit rate is below 10 Mbits/s, response time increases because queuing and processing delays increase (due to the higher volume of background traffic), at a higher rate than the reduction in transmission delays. However, as the link bit rate continues to increase further, the reduction in transmission delays outweighs the increase in processing delays, causing the overall response time to decrease. The reason why the MCS approaches remain reasonably constant (indeed actually decreasing slightly) until the link bit rate increases to 10 Mbits/s is because increasing the volume of background traffic increases the inter-arrival time between foreground packets at the MCS processor since they must queue behind more background traffic throughout the JAMES network. This reduces the queuing delays at MCS processors since only foreground traffic needs to be processed by the MCS.

The response time performance of all type I, II and III messages were similar. This is because all of these messages are similar in size, on the order of 100 bytes. In contrast type IV messages have quite difference performance characteristics due to their larger size. Figure 6.10 compares the mean response time performance of approaches when transmitting large type IV messages. Given the dominant component of large message response times is transmission delay, this figure shows the typical trend of response time decreasing with increasing link bit rate, even when the link utilisation is kept constant.
Figure 6.10: Type IV Message Mean Response Time as Data Link Bit Rate Varies
6. Multicast Delivery

The analysis presented in this section confirms that multicast delivery systems employing either the Cut-Through inter-subnet approach, or the VC Mesh intra-subnet approach provide lower response times than approaches employing the MCS intra-subnet delivery scheme. We had hypothesised that at high link bit rates there would be no significant difference between approaches, however this analysis shows a difference of 20 ms even at a link bit rate of 50Mbits/s. Moreover, the 95th percentile results show that the MCS based approaches are more likely to exceed the type I message delay constraints than the other approaches.

6.3.3 Effect of the Background Traffic Parameters

The previous section indicated that the background traffic characteristics can have a significant impact on the performance achieved for the foreground traffic. To date this chapter has assumed background WWW traffic with an aggregate arrival s.c.v. of 48.1, and average message size of 1169.1 bytes. In practice, background traffic will be a mix of Internet traffic, produced by numerous applications. As stated in [Pax94] due to the variety of Internet applications it is extremely difficult to develop an aggregate Internet traffic model. Hence it is crucial to measure the sensitivity of the alternative multicast delivery approaches to the background traffic characteristics.

Figure 6.11 compares the response time 95th percentile of approaches as the background traffic arrival s.c.v. varies assuming a link bit rate of 5 Mbits/s and a link utilisation of 0.8. This figure shows that all approaches are affected similarly by increasing background arrival s.c.v. As the s.c.v. increases from zero to one hundred the response time 95th percentile of all approaches increases by approximately 130ms. Hence the arrival s.c.v. has a significant impact on the response time performance of all approaches. When the arrival s.c.v. is low, all approaches meet the type I message delay constraint, even though resources have not been reserved for this message flow. However, as the background arrival s.c.v. increases, eventually all approaches are unable to meet the necessary delay constraints. Hence all approaches will require resource reservation techniques to meet the delay constraints in the presence of bursty traffic. The analysis also shows that the Hop-by-Hop approaches are most sensitive to background arrival s.c.v., followed by the Boundary and the Cut-Through approaches.

In terms of s.c.v. values likely to be observed on a network, as discussed above aggregate WWW traffic has an arrival s.c.v. of 48.1. Similarly the arrival s.c.v. of IP datagrams for an FTP connection is around 54 [Pax94]. Moreover, although the aggregate WWW arrival s.c.v. is 48.1, as shown in Section 5.2.2, the arrival s.c.v. of WWW response flows and acknowledgments (in one direction) are 251
Figure 6.11: Type I Message Response Time 95th Percentile as Background Traffic Arrival S.C.V. Varies
6. Multicast Delivery

and 87 respectively. Hence it is possible for background traffic to have an s.c.v. within the range employed in Figure 6.11.

Figure 6.11 also indicates that the Hop-by-Hop, VC Mesh, Source Forwarding Tree approach response time performance matches the Boundary, MCS Source Forwarding Tree approach when the arrival s.c.v. is less than ten, but diverges as the s.c.v. increases further. This is because when the s.c.v. is low, reassembling traffic at routers throughout the JAMES network has a similar impact on total response time as the delay at the subnet MCSs. However as the background traffic becomes burstier, it affects the delay performance of travelling via MCSs more than via routers. This is because traffic must travel through a greater number of nodes when the MCS intra-subnet approach is employed.

The response time performance of all message types followed a similar trend. The response time jitter performance of all multicast delivery systems was also compared. Again, the jitter performance of all approaches were similar. Furthermore, the jitter performance of approaches was not significantly affected by increasing the s.c.v. (decreasing by at most a few milliseconds as the s.c.v. increased to one hundred). That is, the traffic to all receivers is impacted similarly by increasing background s.c.v.

The performance of multicast delivery approaches was also compared as the size of background traffic messages varied. Figure 6.12 shows the response time 95th percentile of approaches as the background message size varies between 48 and 2400 bytes (furthermore, for this scenario, the message size s.c.v. was assumed to be zero). As was the case with varying the arrival s.c.v. this figure shows that the background message size significantly affects the response time performance of all approaches.

When background message size is small (less than 500 bytes) there is a significant difference in the delay performance of approaches employing MCS intra-subnet delivery and all other approaches (on the order of 25 ms). However as the background message size increases further, the response times produced by the Hop-by-Hop VC Mesh and MCS approaches begin to increase more rapidly than the other approaches. Furthermore, the performance of the Boundary MCS and Hop-by-Hop VC Mesh approaches starts to converge. The Boundary VC Mesh and MCS approaches have similar rates of response time increase, since they reassemble traffic at the edge of the core network. However the Boundary VC Mesh approach produces significantly lower delays once the background message size exceeds 1200 bytes.

The Cut-Through approach is least affected by large background traffic messages. This can be seen by the relatively flat curve as the message size in-
Figure 6.12: Type I Message Response Time 95th Percentile as Background Traffic
Message Size Varies
increases above 1200 bytes. Indeed even when the background message size reaches 2400 bytes the Cut-Through approaches still meets the real-time traffic delay constraints, regardless of whether the Source or Shared Forwarding Tree approach is employed.

Hence this analysis shows that there is a significant difference in the sensitivity of approaches to background message size. This is because there is a higher performance penalty on nodes that must reassemble packets as packet size increases than for switches. That is, the difference in approach performance is due to the number of routers each approach employs. As the background message size grows the time to reassemble background traffic increases, in turn increasing the queuing delay for foreground traffic at routers.

Figure 6.12 shows that neither of the Cut-Through approaches exceed the 100ms delay constraint even when the background message size is 2400 bytes. Increasing the background message size further will cause the Cut-Through Shared Forwarding Tree approach to exceed the delay constraint. This is because all foreground traffic must travel via the RP. In contrast, the Cut-Through Source Forwarding Tree approach is relatively insensitive to increasing background message size larger than 1500 bytes. Hence the Cut-Through multicast delivery approaches provides the best response time performance in terms of background message size. This is followed by the Boundary VC Mesh approach. The most important point to note from this analysis is that the delay penalty associated with employing routers (and also MCSs or RPs) within the network, increases as the background traffic message size increases.

6.3.4 Effect of Rendezvous Point Location

The analysis presented in this chapter has shown that there is a significant difference between approaches employing MCS and VC Mesh intra-subnet delivery approaches, yet little difference between those employing Source or Shared Forwarding Trees. Section 6.3.1 hypothesised this was for two reasons: Firstly, the number of links connecting each MCS to the network is much lower than the number of links connecting the RP to the network. This means there is a greater likelihood of getting congestion on the links to each MCS than on the links to the RP. Secondly, the central location of the RP means that most traffic passes through this node regardless of whether the Source or Shared Forwarding Tree approach is employed. Thus there is little delay penalty associated with forcing traffic to travel via the RP. In contrast, traffic only passes through each MCS if the MCS intra-subnet approach is employed. Hence the delay penalty associated with travelling via MCSs can be greater than the penalty associated with travelling via
This section tests these hypotheses by examining the effect of moving the RP from the network centre to each JAMES core network node.

Figure 6.13 compares the response time performance of approaches employing the Shared Forwarding Tree, as the distance between the RP and the network centre increases. Approaches employing the Source Forwarding Tree approach are unaffected by RP location and hence are not shown in Figure 6.13. Figure 6.13 shows that the overall trend is for the response times produced by all Shared Forwarding Tree approaches to increase with increasing RP distance from the network centre. One would expect all approaches to be affected the same by increasing distance since this only increases propagation delay. However, increasing distance and the number of hops tend to be correlated. This can be seen in Figure 6.13 where Hop-by-Hop response times increase with distance more than the other approaches. Furthermore, the response times do not monotonically increase, which is particularly clear when the response time jitters are compared (see Figure 6.14). This 'jagged' behaviour indicates other factors affect response time performance, more than the distance from the network centre. We hypothesise that the number of hops and the number of interfaces to the RP also impact response time per-
6. Multicast Delivery

Figure 6.14: Type I Message Response Time Jitter as RP Distance from Network Centre Varies

Figure 6.15 examines the impact of the number of interfaces to the RP. As hypothesised, response times decrease as the number of interfaces increase. This is because the incoming load to the RP can be spread over multiple incoming and outgoing links, reducing reassembly and transmission queuing delays. The response time graphs in this case are smoother than when the distance to the RP was varied, because the results have been averaged over all locations with the same number of interfaces.

When the number of interfaces are less than five, the Hop-by-Hop VC Mesh and Boundary MCS approaches have similar performance. However as the number of interfaces grows further the Hop-by-Hop VC Mesh approach produces increasingly lower delays than the Boundary MCS approach. This shows that as the number of Shared Forwarding Tree interfaces increases, the choice of intra-subnet approach becomes the dominant contributor to overall delay.

The impact of the number of hops to the RP on response time performance is
Figure 6.15: Type I Message Response Time 95th Percentile as the Number of Interfaces to the RP Varies
shown in Figure 6.16. As expected, the response time achieved by all approaches increases as the number of hops to the RP increases. The number of hops considered is limited, due to the processing requirements required to analyse even moderately sized multicast networks. However, even with the limited size of the network Figure 6.16 clearly indicates the relative sensitivity of approaches to increasing the number of hops to the RP. The rate of increase of the Hop-by-Hop approach response times are the greatest because traffic must be reassembled at each hop. Due to the fact that the Boundary approaches only reassemble at the core network boundaries, increasing the number of hops to the RP, does not increase the frequency of reassembly. Thus the response time performance trend of the Boundary approaches is similar to the Cut-Through approaches. Hence, when there are a large number of hops to the RP, the Boundary and Cut-Through approaches are preferred.

This section has shown that the distance, number of hops and the number of interfaces to the RP do significantly affect the performance of approaches employing Shared Forwarding Trees. Likewise these parameters affect the performance of the MCS intra-subnet approach. If the number of interfaces, or the size of the link to the MCS are increased, the difference between the MCS and VC Mesh approaches becomes insignificant. This analysis emphasises the importance of correctly engineering the network, particularly if Shared Forwarding Trees, or the MCS intra-subnet approach are employed.

6.3.5 Effect of the Number of Senders and Receivers

The analysis presented in this chapter to date has assumed a DIS application involving ten hosts spread over the JAMES network. This section examines the sensitivity of multicast delivery approaches to the number of senders and receivers involved in the multicast communication. Additional hosts (senders or receivers) are assumed to be randomly distributed around the JAMES network.

Figure 6.17 compares the response time performance of multicast delivery approaches as the number of sending DIS hosts varies between 10 and 18. The number of receivers is fixed at 10. This analysis shows that the approaches employing the intra-subnet MCS approach are unable to support greater than 15 senders (when the link to the subnet MCS overloads). In contrast the Cut-Through approaches and approaches employing the VC Mesh intra-subnet approach are able to support 17 senders. The reason why the Shared Forwarding Tree approach is not as sensitive to the number of senders as the MCS approach (even though they both must process traffic from all senders) is because the RP is connected to the rest of the core network via ten links (see Figure 6.3). Hence the traffic
Figure 6.16: Type I Message Response Time 95th Percentile as the Number of Hops between the RP and Network Centre Varies
Figure 6.17: Type I Message Response Time 95th Percentile as the Number of Senders Varies
is distributed over multiple links, rather than being aggregated over one link to local subnet MCSs if the intra-subnet MCS approach is employed. This analysis assumes each additional sender transmits eighty 144 byte messages per second according to a Poisson distribution (i.e. simulates forty DIS ground vehicles as described in Table 6.2). However, similar trends are obtained regardless of the traffic characteristics of the senders added.

This analysis highlights the benefit of the VC Mesh intra-subnet approach, that is to allow traffic to follow the shortest path to the senders, effectively distributing the traffic over the available links. In contrast the intra-subnet MCS based approaches concentrate the traffic from all senders onto a single VC to and from the subnet MCS. Figure 6.17 also shows that as the number of senders increases the difference in response time performance of the Source and Shared Forwarding Trees increases. Again this is because the Shared Forwarding Tree approach forces traffic to concentrate at one point in the network. As discussed earlier the effect of increasing the volume of traffic has less impact on the Shared Forwarding Tree approach than the MCS intra-subnet approach, because the RP is accessible via eleven interfaces compared to one interface to each subnet MCS.

It is also interesting to note from Figure 6.17 that the Cut-Through Source Forwarding Tree approach is relatively unaffected by the increasing number of senders, particularly compared to the other approaches. This is because this approach requires no reassembly between the source and destination. All other approaches require datagrams to be reassembled either at routers, MCSs or the RP. This is emphasised in Figure 6.18 where the response time jitter of approaches are compared. Moreover, Figure 6.18 shows that the higher the frequency of reassembly the greater the magnitude of the jitter and the higher the sensitivity of the approach to the number of senders.

The sender addition analysis shows that the Cut-Through Source Forwarding Tree approach is the least sensitive approach to increasing volumes of traffic of all multicast delivery approaches. Indeed, this approach does not violate the Type 1 message 100 ms delay constraint until the number of senders reaches seventeen. Hence in terms of response time performance the Cut-Through Source Forwarding Tree approach should be selected. The Boundary VC Mesh Source Forwarding Tree approach also performs well, not violating the 100 ms delay constraint until the number of senders reaches fifteen. Thus these approaches are most able to meet the delay requirements of real-time traffic without modifying the best effort behaviour of the Internet.

This analysis clearly indicates that approaches that concentrate traffic, either via the MCS intra-subnet approach or Shared Forwarding Trees are significantly
Figure 6.18: Type I Message Response Time Jitter as the Number of Senders Varies
more sensitive to the number of senders, than approaches that allow traffic to follow the shortest path between the sender and receivers. Therefore, when supporting multicast groups with large numbers of senders, multicast delivery approaches employing the VC Mesh intra-subnet approach and Source Forwarding Trees should be preferred. This is particularly true if the senders are distributed across the network, and hence the path followed by traffic from different senders tends to diverge.

The effect of varying the number of receivers was also examined. However, this analysis found that increasing the number of receivers had little effect on the response time performance of any of the approaches. This is primarily due to the point-to-multipoint delivery of traffic where traffic is only duplicated when the path to two receivers diverges. That is, the load on the links and processors primarily contributing to the response time are not affected by an increase in the number of receivers.

6.4 Conclusions

This chapter has compared the response time performance of ten alternative multicast delivery systems. This analysis has shown that none of the multicast delivery approaches can meet the response time performance requirements of real-time traffic in all network scenarios. Hence mechanisms such as Integrated and Differentiated Services must be employed, particularly if the network is heavily utilised.

Overall the Cut-Through Source Forwarding Tree approach produced the best response time performance in all network scenarios. However all Cut-Through and Boundary VC Mesh approaches provide similar levels of performance in most network scenarios.

In Chapter 3 several issues were highlighted that have not been addressed in the literature. This chapter has investigated all of those issues relating to the delay performance of multicast delivery approaches. The results of our investigation are summarised below.

- Determine in which situations (if any) there is a significant delay difference between the VC Mesh and MCS intra-subnet multicast delivery approaches.

In the majority of network scenarios considered, there is a significant delay difference between the VC Mesh and MCS intra-subnet approaches. The MCS approach consistently produces higher delays than the VC Mesh approach. The main cause of this difference is because the MCS approach concentrates all traffic onto a single link to the subnet MCS. Hence the
response time performance analysis indicates that the VC Mesh approach should be employed. However if the bit rate of the links (or the number of links) connecting the MCS to the remainder of the subnet, are engineered to accommodate the expected total volume of multicast traffic, there is no significant difference between the VC Mesh and MCS approaches.

- Examine the sensitivity of the MCS approach, in terms of delay, to the location of the MCS, and the Shared Forwarding Tree approach to the location of the Rendezvous Point.

Both the MCS approach and the Shared Forwarding Tree approach share the same philosophy of concentrating all traffic to one point (RP or MCS) before forwarding it to multicast receivers. Hence this chapter only considered the RP location, however the analysis is also directly applicable to the MCS approach.

The analysis found that Shared Forwarding Tree (and MCS) approach delay performance is sensitive to the distance, number of hops and number of interfaces to the RP (or MCS). The closer the RP is to the wire-line centre of the network the better the performance of the Shared Forwarding Tree approach. However the sensitivity of multicast delivery approaches to RP location also depended on the inter-subnet approach employed. If the Cut-Through approach is employed delay performance is relatively insensitive to RP location. However the Hop-by-Hop approach is far more sensitive to RP location, since it must reassemble IP datagrams at each network node.

The performance of both the MCS and Shared Forwarding Tree approaches are highly sensitive to the number of interfaces to the MCS or RP. The analysis shows that the smaller the number of interfaces to the MCS or RP the higher the impact on response time performance compared to the VC Mesh and Source Forwarding Tree approaches.

- Investigate whether the Hop-by-Hop inter-subnet delivery approach produces significantly higher delays than the alternative direct VC based inter-subnet approaches.

In moderately sized, lightly loaded networks, where background traffic messages and arrival s.c.v. are small, the Hop-by-Hop approach does not produce significantly higher delays than other approaches. However, the Hop-by-Hop approach is more sensitive to many network parameters than the Boundary and Cut-Through approaches. In particular the Hop-by-Hop approach is sensitive to background message size, number of hops and distance to RP,
and the number of multicast group senders. This means that there are many circumstances where the Hop-by-Hop approach produces significantly higher delays than the other inter-subnet approaches.

- Investigate whether the Boundary variants of the Single Subnet, LSR and NHRP Router approaches have significantly higher delays than the Cut-Through variants of these approaches.

The analysis showed that in the majority of network scenarios there was no significant performance difference between the Cut-Through and Boundary VC Mesh approaches. The Cut-Through approach always produced lower overall delays, but the difference compared to the Boundary VC Mesh approach was only significant when the network was very heavily loaded in some manner (e.g. many senders, low available bit rate, high link utilisation, large background message size). However the Boundary MCS approach produced significantly higher delays than the Cut-Through and Boundary VC Mesh approaches unless there was plenty of capacity on the links to the MCS. Hence either the Cut-Through or Boundary VC Mesh approach can be employed in the majority of multicast environments, with no significant response time performance difference. As discussed above the Boundary MCS approach should only be employed if the links to the MCS are engineered to handle the expected volume of traffic.

- Quantify the delay performance differences between Shared or Source For-warding Trees when employed in an ATM environment.

The Source Forwarding Tree approach always produced lower response times than the Shared Forwarding Tree approach. However in the majority of network scenarios considered the delay performance difference was not significant. The performance of these approaches only differed significantly if the number of interfaces to the RP was small (less than five), or the extra distance, or the number of nodes traversed was high. Hence our analysis shows the importance of selecting the optimal location for the RP. We conclude that in terms of delay either the Source or Shared Forwarding Tree approach can be employed, however the RP location must be selected with care.

- Determine which of the multicast delivery component (intra-subnet, inter-subnet or forwarding tree approach) choices has the greatest impact on the delay performance of the multicast delivery systems.

The analysis presented in this chapter has clearly shown that the choice of intra-subnet approach has the highest impact on delay performance of mul-
Multicast delivery approaches. This was due to the available bit rate connecting the MCSs to their subnets. If the subnets are engineered to alleviate this problem, the choice of inter-subnet approach has the greatest impact on performance, particularly in heavily loaded networks. The choice of Source or Shared Forwarding Tree had no significant impact on overall performance if the RP is located close to the centre of the network, and links connecting the RP to the rest of the network have sufficient capacity.

In general terms the frequency of traffic reassembly (whether at routers, RPs or MCSs) can significantly impact response time performance of multicast delivery approaches. Moreover the impact of reassembly frequency increases with: decreasing available link bit rate, increasing message sizes, and increasing link utilisation. Hence in terms of response time performance, the approaches that reassemble traffic the least provide the best performance.

To conclude our analysis shows, in terms of delay performance, that the Cut-Through, Source Forwarding Tree approach is the optimal multicast delivery system. However in the majority of network scenarios considered there is no significant performance difference if the Boundary, VC Mesh approaches (with either Source or Shared Forwarding Trees) are employed instead of the Cut-Through approach.
Chapter 7

Multicast Delivery Approach

VC Requirements

7.1 Introduction

In the previous chapter the response time performance of alternative multicast delivery approaches were compared. This analysis showed that the Cut-Through Source Forwarding Tree approach produced the best response time performance in all network scenarios. However, as discussed in Chapter 4, when comparing alternative multicast delivery approaches, response time performance is not the only concern. The resource requirements of approaches and their support for dynamic multicast groups must also be considered.

This chapter analyses the VC requirements of alternative multicast delivery approaches. The performance of multicast delivery approaches when supporting dynamic multicast groups, i.e. where senders and receivers can join or leave the group at any time are compared in the next chapter.

As discussed in Chapter 3, one of the key differences between multicast delivery approaches is their VC requirements. The literature recommends that the network should be divided into subnets and the MCS intra-subnet approach employed when VC resources are limited [Arm96, Arm97c]. However, there have been no studies to indicate when the Cut-Through inter-subnet approach and VC Mesh intra-subnet approach VC requirements are so great that they should not be employed. This chapter aims to determine when there are significant difference in the VC requirements of approaches in realistic network scenarios.

The objective of this analysis is to provide recommendations of appropriate multicast delivery approaches based upon: (1) the VC capabilities of the network, and (2) the characteristics of the multicast groups. As outlined in Chapter 3 the
The following performance issues must be addressed.

- Determine in which environments there is a significant difference in the VC requirements of the VC Mesh and MCS approaches.

- Compare the VC requirements of the inter-subnet multicast delivery approaches. In particular there is a need to determine (a) the performance boundaries of each approach, and (b) when there is a significant difference between these approaches.

- Determine the significance of the difference between the VC requirements of the Source and Shared Forwarding Tree multicast delivery approaches.

- Determine which of the multicast delivery component choices has the greatest impact on the VC requirements of the multicast delivery systems.

Section 7.2 compares the VC demands placed on edge devices by each multicast delivery approach. The next section motivates the need to also consider the VC requirements of core network nodes (MCSs, RPs, switches and routers). These devices must support traffic from many different users and hence their VC usage is often even more critical than that of edge devices. Section 7.4 describes the methodology followed to determine the VC demands placed on core network nodes by each multicast delivery approach. In Sections 7.5 to 7.8 approaches are compared in terms of the:

- Number of active links required,
- VCs managed per core network node,
- Subnet VC requirements, and
- Total VC requirements

This chapter then concludes in Section 7.9.

### 7.2 Edge Device VC Requirements

Table 7.1 shows the VC requirements of each approach at senders, receivers, RPs and MCSs, where: \( S \) is the total number of senders, \( S_i \) the number of senders in subnet \( i \), \( N \) the number of subnets containing senders, and \( K \) the number of interfaces that carry traffic from senders to the RP (note \( K \leq N \), since \( N \) subnets contain senders but traffic from more than one subnet may arrive at the RP via the same interface).
As can be seen from Table 7.1, the sender VC requirements are the same for all approaches since senders always transmit only one copy of the multicast traffic. In contrast, the number of VCs terminated by each receiver depends on whether the VC Mesh or MCS intra-subnet approach is employed in Boundary or Hop-by-Hop approaches, or in the Cut-Through case on whether Shared or Source Forwarding Trees are employed. Multicast delivery systems employing the Cut-Through Shared Forwarding Tree approach or the MCS intra-subnet approach terminate one VC per group at receivers, regardless of the number of senders. In contrast, receiver VC requirements for approaches employing the VC Mesh approach depend on the number of senders located in the same subnet. Furthermore, if the Cut-Through Source Forwarding Tree approach is employed, the situation is worse with the number of VCs per receiver depending on the total number of senders in the multicast group.

Hence in terms of VC demands placed on receivers, approaches employing Shared Forwarding Trees or the MCS intra-subnet approach are the best. If the network operator knows the expected number and characteristics of multicast groups, it is straightforward to determine the VC demands placed on receivers for all approaches. For instance, Table 7.2 shows the VC requirements per receiver for each approach as the number of senders varies. This assumes the senders and receivers are uniformly distributed over five subnets, and there are G multicast
groups.

<table>
<thead>
<tr>
<th>Approach</th>
<th>1 Sender (e.g. Video on Demand)</th>
<th>10 Senders (e.g. Audio Conference)</th>
<th>100 Senders (e.g. DIS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-Through, Source</td>
<td>1</td>
<td>10</td>
<td>100</td>
</tr>
<tr>
<td>Cut-Through, Shared</td>
<td>G</td>
<td>G</td>
<td>G</td>
</tr>
<tr>
<td>Hop-by-Hop, Source, VC Mesh Boundary</td>
<td>1</td>
<td>3</td>
<td>21</td>
</tr>
<tr>
<td>Hop-by-Hop, Shared, VC Mesh Boundary</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 7.2: Example of the Number of VCs Terminated by Receivers as the Number of Senders Varies

As the number of senders grows one can see the benefits of (1) sub-dividing the network into subnets and (2) multiplexing traffic via Shared Forwarding Trees or the MCS intra-subnet approach (i.e. rows two and four compared to rows one and three in Table 7.2). Note, these benefits are present, regardless of whether the senders are in one multicast group, or distributed over multiple multicast groups.

Although employing the intra-subnet MCS approach or Cut-Through Shared Forwarding Tree approach reduces the number of VCs terminated by receivers, it shifts the requirement to terminate $S_i + 1$ or $S$ VCs to either the MCS or the RP respectively. Thus in networks where MCSs and RPs have the same VC resource capabilities as receivers, the VC savings of the Shared Forwarding Tree and MCS approaches may not be a significant advantage. There are two counter-arguments to this: (1) there will tend to be fewer MCSs and RPs in the network than receivers, thus the overall VC requirements will be lower, and (2) MCSs and RPs are far more likely to have multiple interfaces connecting them to the network. Typically VCs are limited on a per interface basis rather than over the entire device. Hence, even though the total number of VCs terminated at the MCS or RP may be the same as at receivers in the VC Mesh intra-subnet case, these may be distributed over multiple interfaces. Thus, to dimension networks adequately, operators must compare the VC capabilities of their equipment to the expected multicast group characteristics. For instance, if the VC capabilities of an MCS are not sufficient it may be possible to add an extra interface and thus distribute the VC demands.

One can observe from Table 7.1 that the only difference between the VC requirements of the Hop-by-Hop and Boundary inter-subnet approaches is the num-
ber of VCs the RP must terminate. In the Boundary approach, direct VCs are created across the core network between subnet routers. Hence the RP must terminate, per subnet, one VC for each group containing senders at that subnet. Note, this is in the worst case. It may also be possible for all of the traffic from a given subnet to be multiplexed onto one VC to the RP (regardless of the multicast group it is destined for). The Hop-by-Hop approach terminates VCs at each core network router. This means RPs terminate one VC for each incoming interface carrying traffic from one or more senders if the Hop-by-Hop approach is employed. Knowledge of the network topology is required to determine how this difference impacts the VC requirements of the RP.

This work aims to produce recommendations for operators on the VC demands of alternative approaches in a variety of network topologies. The motivation behind this is presented in the next section.

7.3 Core Network VC Requirements

The worst case VC demands placed on edge devices (senders and receivers) has been considered in the literature [TA97b, TA96b, Arm97b]. However, no analysis has appeared that examines the VC demands placed on routers and switches. The VC requirements placed on these nodes will depend on the network topology. This chapter focuses on the impact of multicast delivery approaches on the VC resources of core network nodes in a variety of network topologies and assuming a variety of multicast group characteristics.

The VC demands placed on core network nodes by multicast delivery systems are particularly of interest to network operators since they tend to process traffic from many different applications, whereas senders and receivers tend to be involved in a much smaller number of applications simultaneously. Thus, for senders and receivers, the size of the multicast group(s) they are involved in is the primary concern. However, for RPs, MCSs, switches and routers, which may be processing traffic from many multicast groups, other factors such as the network topology must be considered. The remainder of this chapter examines the VC requirements of approaches in a variety of networks.

7.4 VC Analysis Methodology

As discussed in the previous section, the VC requirements of core network nodes are highly dependent on the network topology and the characteristics of the multicast groups being supported. For our recommendations to be applicable to a
wide range of operators, the VC requirements of approaches must be compared in a variety of realistic network topologies. This section describes how the network topologies were generated.

To perform this analysis, random networks are designed with Pareto populations and Concave Link Elimination link topologies. Refer to [SEA97] for a more detailed description of this network generation method. This method has been shown to produce low cost, realistic network designs in reasonable computing time [SEA97].

For each network size, we generate thirty different topologies, and determine the VC requirements of each approach on each network. An example ten node network is shown in Figure 7.1. The results are then averaged and 95% confidence intervals obtained.

To compare the VC requirements of the multicast delivery approaches a variable number of multicast groups, senders and receivers can be selected. Table 7.3 shows the range of multicast group parameters considered. Senders and receivers are assumed to be evenly divided between groups and randomly distributed across the network nodes. Each network node represents a single subnet with a topology as presented in Figure 6.2. Furthermore, within subnets, senders and receivers are assumed to be disjoint. This is effectively a worst case scenario in terms of the total number of VCs created within each subnet. As discussed above, it is also assumed that VCs to MCSs and RPs are created on a per group basis rather than traffic from multiple groups sharing one VC. This also represents a worst case scenario.

Table 7.4 shows an example distribution of multicast groups, senders and re-
7. Multicast Delivery Approach VC Requirements

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value Range Considered</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Nodes</td>
<td>3 - 50</td>
</tr>
<tr>
<td>Multicast Groups</td>
<td>1 - 20</td>
</tr>
<tr>
<td>Senders per Group</td>
<td>1 - 100</td>
</tr>
<tr>
<td>Receivers per Group</td>
<td>1 - 100</td>
</tr>
</tbody>
</table>

Table 7.3: Multicast Group Parameter Assumptions

receivers across the network given in Figure 7.1. This example assumes there are five multicast groups, each containing two senders and two receivers.

<table>
<thead>
<tr>
<th>Group</th>
<th>Sender Locations</th>
<th>Receiver Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4, 6</td>
<td>4, 7</td>
</tr>
<tr>
<td>2</td>
<td>0, 8</td>
<td>0, 5</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>2, 3</td>
</tr>
<tr>
<td>4</td>
<td>0, 5</td>
<td>0, 8</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>1, 4</td>
</tr>
</tbody>
</table>

Table 7.4: Example Multicast Group Distribution

In this chapter, the core network and per subnet VC requirements of approaches are examined. Clearly when examining core network VC requirements, approaches that differ only in their intra-subnet approach will have the same VC requirements. For example, in terms of core network statistics, the Boundary, Source Forwarding Tree, VC Mesh and Boundary, Source Forwarding Tree, MCS approaches have the same performance. Therefore, when analysing core network VC requirements only the following distinct multicast delivery approaches need to be considered.

- Cut-Through, Source Forwarding Tree
- Cut-Through, Shared Forwarding Tree
- Hop-by-Hop, Source Forwarding Tree
- Hop-by-Hop, Shared Forwarding Tree
- Boundary, Source Forwarding Tree
- Boundary, Shared Forwarding Tree

As can be seen from Table 7.2, when investigating intra-subnet VC requirements, only the following groups of approaches have different behaviour, and hence need to be considered separately:
• Approaches employing the VC Mesh intra-subnet scheme
• Approaches employing the MCS intra-subnet scheme
• Cut-Through, Source Forwarding Tree approach
• Cut-Through, Shared Forwarding Tree approach

In the remainder of this chapter the VC demands of approaches are compared in terms of their core network VC demands, per subnet VC demands, and overall VC demands. This analysis concludes with several recommendations for network operators indicating, in terms of VC requirements only, which approaches are most applicable in given network and multicast group environments.

As described above, random networks are designed to contain a given number of nodes, multicast groups, senders and receivers. The VC requirements of alternative multicast delivery approaches are then determined in each random network. From this analysis the VC demands of approaches are compared in terms of several core network and subnet VC statistics.

7.5 Number of Active Links Required

Initially the number of active links and number of VCs carried on each link are analysed, for a range of network sizes. We assume five multicast groups, each containing two senders and two receivers. This analysis shows that as the number of network nodes increases, the average number of active links required increases for all approaches (see Figure 7.2). This is because the multicast group members are distributed further apart as the network size increases. This in turn increases the number of links required to connect the multicast group members. All approaches employing the Source Forwarding Tree approach (Cut-Through, Boundary, and Hop-by-Hop) generate the same number of active links since they all create shortest path trees from the senders to receivers. Likewise all Shared Forwarding Tree approaches produce the same number of active links, since they all forward traffic to a RP, which forwards the traffic onto the receivers.

When the network is small all approaches require a similar number of active links. However in larger networks the Source Forwarding Tree approaches require more active links. This is because traffic from each sender follows the shortest path to the destination, whereas in the Shared Forwarding Tree approaches traffic converges at a RP. For instance Figure 7.2 shows that in twenty node networks, the Shared and Source Forwarding Tree approaches employ 15 and 20 active links respectively. However, in fifty node networks their link requirements increase to 27
Figure 7.2: Mean Number of Active Links Required as the Network Size Varies assuming 5 groups with 2 senders and 2 receivers
and 40 respectively. Thus the Source Forwarding Tree distributes the traffic more evenly across the network than the Shared Forwarding Tree approach. This also means that bottlenecks are more likely to occur in the Shared Forwarding Tree approach because traffic from multiple senders follows the same path. However trunking efficiency gains could be made in the Shared Forwarding Tree approach due to its multiplexing nature.

Figure 7.2 shows the 95% confidence interval for each approach. Here we see that although there is a large difference in the mean number of links carrying VCs the confidence interval ranges over-lap. This is because there are some network topologies, where the number of links carrying VCs is similar for both approaches. However in the majority of network topologies considered there was a significant performance difference between the Source and Shared Forwarding Tree approaches in large networks. 95% confidence intervals were calculated for all analysis in this chapter, and similar overlaps were observed as those seen in Figure 7.2. However to aid clarity, graphs only show mean results in the remainder of this chapter.

Figures 7.3 and 7.4 compare the mean number of VCs per link required by each
multicast delivery approach as the network size or number of multicast groups vary. The Hop-by-Hop approaches have the best performance, requiring only one VC per link, regardless of the number of nodes, groups, senders or receivers. This is because traffic is reassembled at each core node, allowing traffic from different senders and even different multicast groups to be carried on the same pt-pt VC between two core nodes. Hence, in terms of VC requirements, the Hop-by-Hop approaches are scalable and place minimal demands on the VC capabilities of core network nodes.

The per link VC requirements of the Cut-Through and Boundary approaches are significantly higher than the Hop-by-Hop approach in small networks. As can be seen from Figure 7.3 the per link VC requirements of these approaches range from 3 to 6 in three node networks. However, their per link VC requirements decrease with increasing network size, with approaches requiring between 1.75 and 2 VCs per link in fifty node networks.

Figures 7.3 and 7.4 show that the Cut-Through and Boundary Shared Forwarding Tree approaches have similar VC requirements per link in large networks, or when the number of senders per group is small. However in small networks, the Cut-Through Shared Forwarding Tree approach has slightly higher VC requirements than the Boundary Shared Forwarding Tree approach in small networks. Figure 7.3 shows this is on the order of 0.5 of a VC in a three node network when groups contain two senders and receivers. However our analysis also showed that the VC requirements difference increased to 15 when each group contains ten senders and receivers). This is because the Cut-Through Shared Forwarding Tree approach requires each sender to create a VC to the RP.

The main reason for the difference is that the Boundary approach can multiplex traffic from senders in the same subnet onto one VC, whereas the Cut-Through approach requires one VC per sender. It is also important to note this analysis assumes that even in the Boundary Shared Forwarding Tree approach, a separate VC is required for each multicast group's traffic. However, since all traffic is travelling to the same RP it could be multiplexed onto one VC. In this case the difference between the Cut-Through and Boundary approaches would be even greater than our analysis indicates.

Figure 7.4 also shows that in multicast environments where the number of senders is large, the Cut-Through approaches can require significantly more VCs per link than the other approaches. For instance, assuming a 5 node network with 5 multicast groups, each containing 100 receivers and senders, the Cut-Through Source and Shared Forwarding Tree approaches required 310 and 120 VCs per link respectively whereas all other approaches require less than twenty VCs per
Figure 7.4: Mean Number of VCs per Active Link as the Number of Groups Vary in 20 Node Networks
7. Multicast Delivery Approach VC Requirements

link. Hence when network operators are supporting multicast groups that contain large numbers of senders that tend to cluster in a few subnets, the Cut-Through approaches should be avoided unless there are other compelling reasons for it being employed (e.g. to meet delay constraints).

Figures 7.3 and 7.4 both show that the Shared Forwarding Tree variants of the Cut-Through and Boundary approaches require more VCs per link than their Source Forwarding Tree counterparts when there are a small number of senders per group. This is because these approaches force traffic to converge onto a smaller number of links because all multicast traffic must travel via the RP. However when groups contain large numbers of senders, the Source Forwarding Tree approaches require more VCs per link. This is because although the Shared Forwarding Tree approach forces all VCs to be carried on the same set of links to the RP, each group only requires one VC between the RP and receivers. This VC saving outweighs the fact that VCs from senders tend to be multiplexed onto fewer links compared to the Source Forwarding Tree approach. Indeed our analysis showed that on average if the group contained three or more senders, the Shared Forwarding Tree approach required less VCs per link than the Source Forwarding Tree approach.

To summarise, this section has highlighted that in terms of the VC requirements per link, the Hop-by-Hop approaches are superior, requiring at most one VC per link, regardless of network size, number of multicast groups, senders or receivers. The analysis has also shown that the Boundary Source and Shared Forwarding Tree approaches provide the next best performance because they multiplex traffic from multiple senders located in the same subnet onto one VC per group. Thus, the Boundary approaches can perform significantly better than the Cut-Through approaches, especially in small networks (i.e. where multiple senders cluster in each subnet), or when there are a large number of senders.

This section has also indicated that if multicast groups contain three or more senders, the Source Forwarding Tree approaches require more VCs per link than the Shared Forwarding Tree approaches. If multicast groups contain fewer senders, the Source Forwarding Tree approach requires fewer VCs per link. This is because links to the RP tend to support a higher number of VCs than links in the Source Forwarding Tree approach, because all multicast traffic must converge on these links. However, links from the RP to receivers only require one VC per group. Hence, in networks where the number of senders is small and the number of receivers is large, the difference in the VC requirements between Source and Shared Forwarding Tree approaches can become insignificant.
7. Multicast Delivery Approach VC Requirements

7.6 VCs Managed per Core Node

This section examines the number of VCs managed by core network nodes. This is an important statistic because it impacts the processing requirements, signalling resources, and table storage requirements of core nodes. Furthermore, initiating or terminating VCs at a node implies the use of segmentation and reassembly, adding both to processing demands and end-to-end delay.

The number of VCs managed per core node depends on the intra-subnet multicast delivery approach employed. For instance, if the VC Mesh approach is followed a core node must terminate one VC for each sender in its subnet. In contrast if the MCS approach is employed, the core node only needs to terminate one VC for each group represented in its subnet. Thus this section compares the VC requirements of all ten multicast delivery approaches. First we consider VC requirements when the multicast group characteristics are fixed and the network size is varied. Then we consider the impact of multicast group characteristics on core node VC requirements.

Figure 7.5 confirms, as expected, that the number of VCs managed per core node decreases with increasing network size for all approaches. As discussed earlier, this is because the multicast group members are distributed farther apart as the network size increases. This analysis indicates that when the network contains fewer than 25 nodes, the Boundary Source Forwarding Tree approaches require core nodes to manage more VCs than other approaches employing Source Forwarding Trees. In the Boundary approaches, each core node must terminate VCs from all senders in its local subnet if the VC Mesh approach is employed, or one VC per group if the MCS approach is employed. The core node must also create/terminate one pt-mpt VC across the core network for each group that has senders/receivers located in that core node’s local subnet.

The Cut-Through approach places the lowest VC demands on core nodes. This is because the Cut-Through approach creates VCs end-to-end which means each core node has to manage at most one VC per sender. Thus this analysis highlights one of the disadvantages of reassembling traffic at the edge of subnets. That is, it places a higher VC management burden on core nodes because they must manage VCs within the core network and VCs within the local subnet. When the network supports five multicast groups (each with two senders and two receivers), the Cut-Through approach requires core nodes to manage between 25 and 50% less VCs than the Boundary approach. However, when either the number of multicast groups, or number of senders in each multicast group increases, the aggregation benefits of the Hop-by-Hop and Boundary approaches outweigh the benefits of
Figure 7.5: Mean Number of VCs Managed per Core Node as Network Size Varies assuming 5 groups with 2 senders and 2 receivers
having direct end-end VCs. In these environments the Cut-Through approach requires core network nodes to manage more VCs than all other approaches.

When there are a small number of small multicast groups (see Figure 7.5), the performance of the Hop-by-Hop approaches falls between the Boundary and Cut-Through approaches. This is because the Hop-by-Hop core nodes have to manage the same number of VCs local to the subnet as the corresponding Boundary approaches. However, the Hop-by-Hop approach core nodes manage at most one VC per core network interface. The difference between the Hop-by-Hop and Boundary approaches is most significant when the network is small, and the difference decreases with increasing network size. The reason for this is that in small networks senders from multiple groups tend to be located in the same subnet. This means that the Boundary approach core nodes will have to manage a large number of VCs within the core network, In contrast, the Hop-by-Hop node manages at most one VC per core network interface. This difference is most pronounced when the Shared Forwarding Tree approach is used. This is because the results are skewed by the number of VCs the RP must manage. In the Boundary approach the RP must manage one outgoing VC per group in addition to one VC per group per sending subnet. In contrast since the Hop-by-Hop approach employs pt-pt (bi-directional) VCs, the RP must manage at most one VC per interface.

Indeed, in the Shared Forwarding Tree case, the Hop-by-Hop VC Mesh approach requires core nodes to manage less VCs than the Cut-Through approach, even when there are only two senders per group. This is because the Cut-Through Shared Forwarding Tree approach requires the RP to terminate one VC per sender and initiate one VC per group. In contrast the Hop-by-Hop approach requires the RP to terminate at most one VC per interface. As the number of network nodes increases, the influence of the RP decreases to the extent that in fifty node networks the Cut-Through approach has lower VC management requirements than the Hop-by-Hop VC Mesh Shared Forwarding Tree approach. Our analysis also showed that increasing either the number of multicast groups, or the number of senders in each group, had the same impact on the relative performance of the Hop-by-Hop and Cut-Through approaches.

Figure 7.5 also shows that when multicast groups contain two senders and two receivers, the MCS intra-subnet approach requires core nodes to manage more VCs than the VC Mesh approach. This is contrary to the commonly held hypothesis that the MCS approach has lower VC requirements than the VC Mesh approach. The reason for this is that the MCS approach reflects traffic back to senders if they are also receivers, as discussed in Chapter 3. Hence if a multicast group has receivers within and external to a core node's local subnet, the router will
receive reflected copies of packets it transmits into the subnet. This will occur even when the subnet does not contain any other active senders unless the MCS is enhanced, so that it does not add a receiver to the pt-mpt VC if that node is also the only active sender in the subnet. Our analysis showed that even when the network supports 20 groups each with ten senders and receivers, the VC Mesh approach out-performs the MCS approach, if the network contains more than 15 nodes. Once the number of senders per group was greater than 20, the MCS intra-subnet approach out-performed the VC Mesh approach, regardless of the number of groups.

Figure 7.6 shows the impact of increasing the number of groups, senders and receivers on approaches employing Source Forwarding Trees, assuming 20 node networks. Similar results were obtained when the Shared Forwarding Tree approach was employed. As can be seen from Figure 7.6, the Boundary and Cut-Through approaches are affected more by increasing the number of groups than the Hop-by-Hop approaches. The reason why the Boundary approaches are affected is that core nodes create VCs on a per group basis. Moreover, in our analysis increasing the number of groups, also increases the number of senders (since we assume two senders and receivers per group). This is why the Cut-Through approach is also significantly impacted by the number of groups.

Figure 7.6(c) also shows the impact of increasing the number of senders. This clearly indicates that the Boundary and Cut-Through approaches are far more sensitive to the number of senders than the Hop-by-Hop approaches. We found this was particularly true when Source Forwarding Tree approaches are employed. For instance, when there are twenty groups, each with ten senders and two receivers, our analysis showed that core nodes manage on average 57 and 43 VCs for the Source and Shared Forwarding Tree variants of the Boundary approach respectively. Furthermore, Figure 7.7 shows that unless multicast groups contain more than ten senders, the Cut-Through approach requires core nodes to manage less VCs than the Boundary approach. This is because the Boundary approach terminates VCs at the edge of the core network whereas the Cut-Through approach does not.

To summarise, in networks supporting large multicast groups the Boundary approaches place the highest VC requirements on core nodes. However, in networks with large numbers of senders, the Cut-Through approach will place the highest demands on the core nodes. Figure 7.7 shows that once the number of senders per group increases above ten, the Cut-Through approach places significantly higher VC requirements on core nodes than the other approaches. However, we also found that the number of VCs managed by Cut-Through approach core
7. Multicast Delivery Approach VC Requirements

Figure 7.6: Mean Number of VCs Managed per Core Node in Source Forwarding Trees as the Number of Groups Vary in 20 Node Networks

(a) 2 Senders and 2 Receivers

(b) 2 Senders and 10 Receivers

(c) 10 Senders and 2 Receivers
Figure 7.7: Mean Number of VCs managed per Core Node as the Number of Senders vary, assuming 20 node networks containing 5 groups each with 2 receivers - Source Forwarding Trees
nodes halves when a Shared Forwarding Tree is employed.

This section has focused on the number of VCs core nodes must manage when each multicast delivery approach is employed. This influences the signalling, data processing and memory requirements of core nodes. The analysis has clearly shown that the Hop-by-Hop approaches place the least demands on core nodes even though VCs are terminated and initiated at each core node. This is because the multiplexing benefits of the Hop-by-Hop approach outweighs the fact routers must manage VCs within the subnet and within the core network. Furthermore, the Hop-by-Hop approach has the advantage that it can employ bi-directional pt-pt VCs so traffic flowing in both directions can be supported by a single VC. In contrast the Boundary and Cut-Through approaches employ a large number of pt-mpt VCs which are unidirectional. The benefits of the Hop-by-Hop approach are particularly noticeable in small networks, or networks containing large numbers of groups, senders or receivers. The only situation where the Cut-Through approach out-performs the Hop-by-Hop approach is when the network supports a small number of groups each with only two senders. However, the Cut-Through approach requires core nodes to manage less VCs than the Boundary approach, unless multicast groups contain more than ten senders.

7.7 Subnet VC Requirements

The initial part of this chapter has focused on VC requirements within the core network. This section analyses the VC demands of alternative approaches within subnets. Table 7.1 showed the VC demands per receiver, sender and MCS. Hence in this section we focus on the total number of VCs initiated or terminated within the subnet. This will impact the number of VCs the edge router or switch must manage and also indicate the VC demands placed on local subnet switches. This chapter has assumed a simple subnet, where all senders and receivers are connected by a single switch which must support all VCs within the subnet.

Initially the number of VCs initiated within a subnet is examined. Note the forwarding techniques employed in the core network will have no impact on the number of VCs initiated in the local subnet. Therefore the Boundary and Hop-by-Hop Source and Shared Forwarding Tree approaches will all initiate the same number of VCs within a subnet. Hence only the Cut-Through Source and Shared Forwarding Tree approaches, the VC Mesh approach and the MCS approach need to be considered separately.

Figure 7.8 shows that the MCS approach initiates the highest number of VCs per subnet: 7.75 compared to 4 for the VC Mesh approach and 2 for the two Cut-
Figure 7.8: Mean Number of VCs Initiated per Subnet, as the Number of Groups Varies assuming 2 senders and 2 receivers per group and 20 node networks

Through approaches. Similar differences were observed when there are only five groups when the network contained five nodes rather than twenty (10, 6 and 3.5 VCs for the MCS, VC Mesh, and Cut-Through approaches respectively). We also observed that the number of VCs initiated, decreased with increasing network size due to the increasing distribution of senders and receivers across multiple subnets. That is, the difference between approaches is most significant when senders and receivers are clustered into a small number of subnets, or when the network supports many multicast groups.

The two Cut-Through approaches are affected the least by the number of groups, increasing from half a VC being initiated in each subnet to two VCs initiated per subnet as the number of groups varies from one to twenty. In both Cut-Through approaches the only parameter that impacts the number of VCs initiated is the number of senders in each subnet. Given each new group contains two new senders the number of senders per subnet will gradually increase. Note, that if each group contained more senders this would have a greater impact on the Cut-Through approaches. However the same VC demands could be created in a multicast environment containing a small number of groups with many senders.
rather than a large number of groups with small numbers of senders. In contrast the VC Mesh and MCS approaches are more sensitive to the number of groups, than the number of senders within each group.

The VC Mesh approach has the next best performance initiating on average four VCs per subnet when the network supports twenty groups. The numbers are higher than the Cut-Through approaches because the edge router must create a VC for each group that has receiver’s in its subnet. When the network supports twenty groups this will mean there are forty senders and receivers, and hence on average two senders and receivers per subnet. This is why the mean number of VCs initiated is around two for the Cut-Through approaches and four for the VC Mesh approaches.

The number of VCs initiated in the MCS approach is much higher (just below eight VCs initiated when there are twenty groups). If the subnet contains two senders and two receivers the MCS will need to initiate four VCS if the senders and receivers are all in different groups, i.e. the MCS must initiate one VC per group. This will impact both the MCS and the edge router because the edge router will be a leaf on each of the distribution VCs created by the MCS so that it can forward traffic into the core network. Hence, when the Boundary or Hop-by-Hop approaches are employed in networks with large numbers of small groups, greater demands are placed on the VC capabilities of each core node (and the subnet switches) if the MCS intra-subnet approach is employed because the router must terminate one VC for each group represented in the subnet.

It is also important to note that although the total number of VCs initiated in a subnet will always be higher when the MCS intra-subnet approach is employed, if the subnet contains a large number of senders which are all members of the same groups, the number of VCs the edge router has to manage will be smaller in the MCS approach compared to the VC Mesh approach. This is because the edge router only needs to manage one VC per group with local senders and VCs for traffic arriving from the core network, rather than managing one VC per local sender.

When examining the number of VCs terminated in each subnet, approaches employing Source and Shared Forwarding Tree approaches need to be considered separately. This is because the Forwarding Tree approach affects the number of VCs terminated at the router on the edge of each subnet in the Hop-by-Hop and Boundary approaches. As can be seen from Figure 7.9, the relative performance of approaches can differ significantly on the basis of whether Source or Shared Forwarding trees are employed.

In the case of Source Forwarding Trees, the Hop-by-Hop and Boundary VC
7. Multicast Delivery Approach VC Requirements

Mesh approaches terminate the lowest number of VCs in each subnet: one VC per active group. Regardless of the number of senders or receivers, or the forwarding tree approach employed, the Hop-by-Hop and Boundary MCS approaches terminate the highest number of VCs per subnet. In these cases, one VC is terminated per local sender, and either one or two per active group in that subnet (if that group has external senders there will be two VCs terminated for that group, one between the router and MCS and one between the MCS and local receivers).

The number of VCs terminated by the Cut-Through approach within a subnet depends on the number of senders and receivers. When each group contains a small number of senders and receivers, the Cut-Through Source Forwarding Tree approach performs like the VC Mesh approaches, approximately terminating one VC per active group. However, when the number of senders or receivers in each group increases, the Cut-Through Source Forwarding Tree requires significantly more VCs to be terminated. Our analysis showed that even when groups contain ten senders, the Cut-Through approach requires fewer VCs to be terminated per subnet than the MCS approach. However, when the multicast groups contain more than ten senders, the Cut-Through Source Forwarding Tree approach requires each subnet to terminate more VCs than any other approach. This is because the Cut-Through Source Forwarding Tree approach terminates VCs on a per sender basis rather than a per group basis.

When multicast delivery approaches employ Shared Forwarding Trees, the Cut-Through approach terminates the lowest number of VCs per subnet regardless of the number of senders or receivers. This is because in this case, traffic from all senders is multiplexed onto a single VC per group at the RP. Hence each subnet only needs to terminate at most one VC per active group. In contrast, both the Hop-by-Hop and Boundary approaches create VCs from local senders which terminate within the subnet even when the core network employs the Shared Forwarding Tree approach. This shows the multiplexing benefit of converging all traffic at one central RP, rather than distributed multiplexing at local subnet MCSs. The disadvantage is that all multicast traffic must travel via the RP, even if the senders and receivers are located in the same subnet, causing higher traffic load (if all senders and receivers are local), and increasing end-end delay.

7.8 Total VC Requirements

This chapter concludes with an analysis of the total VC requirements of all approaches. Figure 7.10 compares the total number of VCs required by all approaches incorporating both their core network and subnet requirements, assuming Source
Figure 7.9: Mean Number of VCs Terminated per Subnet as the Number of Groups Vary in 20 Node Networks with 2 Senders and 10 Receivers per Group
7. Multicast Delivery Approach VC Requirements

(a) Source Forwarding Trees

Figure 7.10: Total VC Requirements Assuming Source Forwarding Trees and 5 groups each with 2 senders and 2 receivers

Forwarding Trees. Similar results are achieved if Shared Forwarding Trees are employed, the main difference being the magnitude of VC requirements is higher in the Shared Forwarding case, due to the need to terminate and initiate new VCs at the RP.

The total number of VCs required by the Cut-Through Source Forwarding Tree approach depends only on the number of senders since one pt-mpt VC is created per sender. Similarly, the Cut-Through Shared Forwarding Tree approach total VC requirements depend only on the number of senders and the number of multicast groups. The VC requirements are sensitive to the number of groups because the RP must create a pt-mpt forwarding VC for each multicast group. Hence, one advantage of the Cut-Through approaches is that they are insensitive to the size of the network. However it is important to note although the overall VC requirements of the Cut-Through approach are lower than for other inter-subnet delivery approaches, the demands on individual links and core network nodes are far greater than for other approaches.

The Boundary approaches are also relatively unaffected by network size, par-
particularly if the network contains more than ten nodes. The Boundary approaches create forwarding VCs on a per subnet and per multicast group basis, rather than on a per sender basis. Hence when the network is small, and multiple senders tend to be located in one subnet, the total VC requirements of the Boundary approaches are lower than in larger networks where only one sender or receiver tends to be located at a node.

The Hop-by-Hop approaches are most sensitive to the size of the network, because they create VCs on a link by link basis. Hence the total VC requirements of the Hop-by-Hop approaches increase with increasing network size. However, as discussed earlier, the primary benefit of the Hop-by-Hop approach is that although it requires the highest number of VCs in total, at most one pt-pt VC is required on each link. In contrast all Cut-Through and Boundary approaches may require multiple VCs per link. Given, most operators are constrained by the number of VCs managed by a single node rather than the total VC requirements, in terms of VC characteristics the Hop-by-Hop approaches provide the best performance. The trade-off is that the Hop-by-Hop approach requires data to be reassembled at each core node, increasing forwarding delay compared to the Cut-Through and Boundary approaches.

The difference between the Hop-by-Hop Source and Shared Forwarding Tree approaches is whether or not traffic is forwarded to a RP before being forwarded to receivers. There is no advantage in terms of a VC reduction per link, because regardless of the forwarding tree technique employed, Hop-by-Hop approaches require at most one VC per link. The advantage of Source Forwarding Trees is that they reduce end-end delay because all traffic can follow the shortest path from senders to receivers. Hence the decision as to whether the Source or Shared Hop-by-Hop approach should be employed depends on whether the operator is more concerned about limiting the number of links carrying multicast traffic or latency.

The impact of the number of groups on total VC requirements was also analysed. Approaches employing the MCS intra-subnet approach have the highest VC requirements in networks with many groups, particularly if operated in conjunction with the Boundary approach. This is because the Boundary approach creates VCs both within subnets and across the core network on a per group basis. The Cut-Through approaches are least sensitive to the number of groups, particularly when the Source Forwarding Tree approach is employed.
7. Multicast Delivery Approach VC Requirements

7.9 Conclusions

The objective of this chapter was to provide recommendations of appropriate multicast delivery approaches based upon: (1) the VC capabilities of the network, and (2) the characteristics of the multicast groups. This section highlights our findings for each of the performance issues outlined in Section 7.1.

- Determine in which environments there is a significant difference in the VC requirements of the VC Mesh and MCS approaches.

When considering the number of VCs that have to be managed within a subnet there is a significant difference between the VC Mesh and MCS approaches, particularly when the subnet must support many multicast groups. The MCS approach will always require receivers and routers to terminate less VCs than the VC Mesh approach. However when the subnet contains senders or receivers that belong to many different groups the overall VC requirements, and thus the number of VCs the subnet switches have to manage, are much higher when the MCS approach is employed. This is particularly true when the subnet must support many multicast groups. The benefit of the MCS approach comes into its own when the subnet must support a small number of groups with a large number of local senders. However if the subnets supports many multicast groups, each with only one or two local senders the VC Mesh approach is preferable.

This indicates that operators may wish to employ a dynamic hybrid approach where the VC Mesh approach is employed for each group that has one or two local senders. However if a multicast group contains more than two local senders, the MCS approach is employed. Thus at any one time some groups may be supported via the VC Mesh approach, and others via the MCS approach.

- Compare the VC requirements of the inter-subnet multicast delivery approaches. In particular there is a need to determine (a) the performance boundaries of each approach, and (b) when there is a significant difference between these approaches.

In terms of VC requirements the Hop-by-Hop approach is the most scalable. Regardless of the number of groups, senders, or receivers it requires at most one VC per link. Furthermore, due to their pt-pt nature, the same VCs can also be reused for unicast traffic. If the Hop-by-Hop approach is employed, core nodes at the edge of subnets must manage VCs within their local subnet and at most one VC per core network interface. However pure core network
nodes (those not directly attached to a subnet) have bounded VC requirements of at most one VC per core network interface. The number of VCs per link is also bounded to one. This is true for best effort traffic. However for real-time traffic multiple VCs may need to be created each with different QoS characteristics. Depending on how many different QoS classes need to be supported the difference between the Hop-by-Hop and other inter-subnet approaches will start to decrease.

Our analysis showed that there is a trade-off between the Cut-Through and Boundary approaches. When multicast groups contain one or two senders the Cut-Through approaches are superior because there are no multiplexing gains to be made at the edge of subnets. Moreover core nodes at the edge of subnets manage less VCs with the Cut-Through approach. This is because it employs end-end VCs rather than terminating VCs at the edge of subnets as in the Boundary approach. However, when multicast groups contain many senders, particularly if they are clustered in a small number of subnets, the Boundary approach has better VC characteristics within the core network, since traffic is multiplexed onto one VC per group at the edge of each subnet. Given core network nodes have to support many multicast groups simultaneously, the VC demands placed on these nodes will often be of more concern to operators. Thus, even though the Boundary approaches require more VCs to be managed within subnets and at the edge of subnets, in terms of the core network they are preferable.

In general, the Cut-Through approaches are extremely sensitive to the number of senders, whereas the Boundary approaches are more sensitive to the number of multicast groups. Hence in cases where there are many more senders than groups the Boundary approaches are better than the Cut-Through approaches. However, when the number of groups and senders are similar, the Cut-Through approach is preferable because it reduces the VC demands within subnets and on core nodes directly attached to these subnets. The other benefit of Boundary approaches compared to Cut-Through approaches is that they can be employed by core network operators, regardless of the link layer technology employed in subnets (e.g. ATM, Gigabit Ethernet etc). In contrast, the Cut-Through approaches assume end-end ATM connectivity.

To summarise, for transit core network nodes the Hop-by-Hop approach is far superior to all other multicast delivery approaches in terms of VC requirements. The disadvantage of the Hop-by-Hop approach, as seen in
Chapter 6, is that multicast traffic must be reassembled at each core network node increasing: (1) the processing load on these nodes, and (2) end-end delay. When these factors are an issue the Cut-Through approach should be employed when multicast groups only contain one or two senders. In all other situations the Boundary approach is preferable to the Cut-Through Approach.

- Determine the significance of the difference between the VC requirements of the Source and Shared Forwarding Tree multicast delivery approaches.

The advantage of Shared Forwarding Trees is that they multiplex traffic from multiple senders onto one forwarding VC per group. This reduces the number of VCs core nodes between the RP and receivers must terminate. However when multicast groups contain one or two senders, the Source Forwarding Tree approach requires less VCs per link than the Shared Forwarding Tree approach. Furthermore this approach distributes the VCs across the network more evenly. However when multicast groups contain three or more senders (regardless of the number of receivers), employing the Shared Forwarding Tree approach requires less VCs per link than the Source Forwarding Tree approach.

The other disadvantage of the Shared Forwarding Tree approach is that the central multiplexing node (RP) must manage a large number of VCs, up to one per sender, plus one per group if employed in conjunction with the Cut-Through approach. The trade-off is the reduction in the number of VCs core nodes between the RP and receivers must terminate.

If forwarding tree approaches are employed in conjunction with the Cut-Through inter-subnet approach, the Shared Forwarding Tree should be employed. This is because it places much lower demands on the subnet switches and receivers than the Source Forwarding Tree approach. If the Boundary approach is employed, the Shared Forwarding Tree approach only offers significant VC savings if multicast groups contain many senders located in different subnets. When the Hop-by-Hop approach is employed, there is no difference between the Source and Shared Forwarding Tree approaches in terms of the VC requirements per link. However, since the Shared Forwarding Tree approach forces all traffic to travel via the RP, it will reduce the total number of links that must support multicast VCs.

- Determine which of the multicast delivery component choices has the greatest impact on the VC requirements of the multicast delivery systems.
The choice of inter-subnet multicast delivery approach has the greatest impact on overall multicast delivery system VC requirements. If the Hop-by-Hop approach is employed, regardless of the intra-subnet or forwarding tree technique required, at most one VC must be managed per core network interface. The characteristics of the multicast groups being supported also has the greatest impact on the inter-subnet approach employed. If multicast groups contain small numbers of senders there is little difference between the Boundary and Cut-Through approaches. However there is a large difference between these approaches when groups contain many senders. The impact of the intra-subnet approach tends to be less significant. There only tends to be a significant difference between the VC Mesh or MCS approaches when a subnet contains large numbers of senders. The forwarding tree approach also had little impact on the relative performance of overall multicast delivery systems, unless employed in conjunction with the Cut-Through approach.

To conclude this chapter has clearly shown the benefits of the Hop-by-Hop approach in terms of VC requirements, compared to the Cut-Through and Boundary approaches. The analysis has also highlighted the value of multiplexing traffic at RPs or local MCSs in terms of reducing VC requirements on receivers. If either of these approaches are employed, receivers only need to terminate one VC per multicast group they join. In contrast other approaches require them to terminate up to one VC per sender (in the case of the Cut-Through Source approach). Hence in environments where multicast groups contain many senders, Shared Forwarding Trees and or the MCS intra-subnet approach should be used. The disadvantage of multiplexing traffic is the number of VCs the MCSs or RPs must manage. However, these nodes tend to have multiple interfaces connecting them to the network, compared to receivers and thus the VC requirements can be amortised over multiple interfaces.
Chapter 8

Multicast Dynamic Group Analysis

8.1 Introduction

In the previous two chapters the response time performance and VC requirements of alternative multicast delivery approaches were compared. This showed that the Cut-Through Source Forwarding Tree approach produced the best response time performance in all network scenarios. Furthermore, the Hop-by-Hop approach had the lowest VC requirements. However, Chapter 6 illustrated that the Hop-by-Hop approach provides the worst response time performance of all inter-subnet delivery approaches. Thus there is a trade-off between response time performance and VC requirements.

Chapter 6 assumed that multicast group membership is static. However in practice many multicast applications, including DIS, have dynamic multicast group membership, i.e. where senders and receivers can join or leave the group at any time. Moreover, Sections 3.3 and 3.4 highlighted that one of the major differences between multicast delivery approaches is how they perform when supporting dynamic multicast groups, both in terms of delay and signalling resource requirements. Hence it is critical that the performance of alternative multicast delivery systems are compared when supporting dynamic multicast groups. This chapter investigates the performance of approaches in terms of: (a) the time to add or remove senders and receivers, and (b) the signalling network resources required to add or remove senders and receivers.

Section 8.2 describes the alternative approaches for supporting dynamic multicast groups. The methodology employed to compare the approaches is described in Section 8.3. Approach delay performance (i.e. the time to add senders or
receivers) is analysed in Section 8.4. This is followed by a comparison of the signalling resource requirements of each approach in Section 8.4.4. This chapter concludes in Section 8.5.

8.2 Approaches for Dynamic Multicast Group Support

In Chapters 6 and 7, ten multicast delivery approaches were compared, employing a combination of the inter-subnet, forwarding tree, and intra-subnet approaches shown in Table 8.1.

<table>
<thead>
<tr>
<th>Scope</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inter-Subnet</td>
<td>Cut-Through</td>
</tr>
<tr>
<td></td>
<td>Boundary</td>
</tr>
<tr>
<td></td>
<td>Hop-by-Hop</td>
</tr>
<tr>
<td>Forwarding Tree</td>
<td>Source</td>
</tr>
<tr>
<td></td>
<td>Shared</td>
</tr>
<tr>
<td>Intra-Subnet</td>
<td>VC Mesh</td>
</tr>
<tr>
<td></td>
<td>MCS</td>
</tr>
</tbody>
</table>

Table 8.1: Multicast Delivery Approaches

In Chapters 6 and 7, the two variants of the Single Subnet, LSR, and NHRP Router approaches were grouped together as the Cut-Through and Boundary approaches. This is because all three approaches forward traffic across the network in the same way. However, these three approaches create VCs differently and hence should be considered separately when analysing dynamic multicast group support. In this chapter, we only consider the performance of traditional ATM signalling techniques. Thus only the Single Subnet (SS) approach is analysed. The LSR and NHRP Router approaches were defined with the premise that the traditional ATM Single Subnet approach performance was inadequate in terms of delay and signalling resources needed to create the VCs. This chapter will explore this hypothesis to determine if it is true, and leave the detailed analysis of the LSR and NHRP Router approaches for further study. Hence this chapter compares the Hop-by-Hop, Cut-Through - Single Subnet, and Boundary - Single Subnet inter-subnet multicast delivery approaches. Note also that the difference between the Cut-Through and Boundary modes of approaches is whether the direct VC extends across the entire ATM network, or only within the core network, with reassembly occurring on subnet boundaries.

Chapter 4 presented the performance measures that must be considered when adding or removing senders, or receivers, from a multicast group. A user is most
8. Multicast Dynamic Group Analysis

concerned about the time it takes for (a) data from a new sender to reach receivers, and (b) data from senders to reach a new receiver. Hence both of these measures are analysed in this chapter. A network provider is concerned about the resources required to add, or remove, a new sender or receiver. This is primarily the signalling processor resources required to complete the join or leave dynamic group process. The signalling resource requirements of each approach are also compared in this chapter. A summary of the chapter objectives is listed below:

- Determine if there is a significant difference in the time it takes the MCS and VC Mesh approaches to add new (or remove) senders or receivers.
- Investigate whether the time to add new senders or receivers significantly differs between the alternative inter-subnet multicast delivery approaches.
- Examine the difference between Source and Shared Forwarding Trees when supporting dynamic multicast groups.
- Determine the relative signalling network demands of the alternative intra-subnet, inter-subnet and forwarding tree approaches.
- Determine which of the multicast delivery component choices has the greatest impact on the dynamic group support performance of the multicast delivery systems.

8.3 Dynamic Group Analysis Methodology

Regardless of the multicast delivery approach employed, multicast group management information must be maintained. That is, a multicast group management protocol is required to enable new senders, existing senders, subnet MCSs, RPs and routers to add a new sender or receiver to a multicast group. Within a subnet, this thesis employs the MARS multicast group management protocol [Arm96] since this is the only standardised approach for supporting IP multicast groups over ATM networks. The Hop-by-Hop approach employs an Internet multicast routing protocol (we assume PIM-SM) to create Source and Shared Forwarding trees.

Although the Single Subnet approaches also create either Source or Shared Forwarding Trees they do not use Internet multicast routing protocols such as PIM-SM to achieve this, since the network nodes are ATM switches, not IP routers. Hence the entire network (or in the Boundary case, the core network) is treated as one subnet. Thus the MARS VC Mesh and MCS intra-subnet approaches are
also employed to create the Single Subnet Source and Shared Forwarding Trees respectively.

The MARS protocol deploys a Multicast Address Resolution Server (MARS) which maintains the ATM addresses of multicast group members in the local subnet. Potential senders and receivers must register with the MARS to receive group membership information. MCSs must also register the multicast groups they are willing to support. When a new sender or receiver wishes to join the multicast group, it must inform the MARS. In the case of a new sender, the MARS will return a list of the addresses of all receivers. The new sender uses this information to create a pt-mpt VC to all receivers. In the case of a new receiver, the MARS will inform all of the local senders (or the MCS) of the new receiver’s ATM address. Each sender (or the MCS) will add the new receiver to their pt-mpt VCs. For more details of the MARS protocol refer to [Arm96].

The Hop-by-Hop approach employs the MARS protocol within each subnet in conjunction with either the VC Mesh or MCS approach. The Cut-Through Single Subnet approach treats the entire network as one subnet, hence the MARS VC Mesh and MCS approaches are employed over the entire network. The Boundary Single Subnet approach breaks the network into subnets and the core network. The MARS approaches are employed within each subnet, and to manage the multicast VCs across the core network.

Figures 8.1 and 8.2 show the MARS message flows required to add a new sender or receiver to a multicast group, assuming there are already multicast group members in that subnet. Note the message passing observed by senders and receivers is identical for both the VC Mesh and MCS approach. The only difference as far as senders and receivers are concerned is the list of addresses contained in each message (the VC Mesh address list will contain all local receivers and the subnet router, whereas the MCS address list will contain the local MCS address only). Although there is no difference in the MARS message passing (other than message sizes), Figures 8.1 and 8.2 show that the VC Mesh approach generates more signalling traffic than the MCS approach. In this chapter we will investigate the significance of this difference.

The difference between inter-subnet approach (Hop-by-Hop, Boundary Single Subnet and Cut-Through Single Subnet) dynamic multicast group support is the scope of the VCs being modified. In the Hop-by-Hop and Boundary approaches two scenarios must be considered: the first is when the host (sender or receiver) joining or leaving a group is attached to a subnet that contains other multicast group members. The second case is when the host is the first host in its subnet to join (or leave) that multicast group.
8. Multicast Dynamic Group Analysis

Figure 8.1: Message Flows to Add New Senders

Figure 8.2: Message Flows to Add New Receivers
When the subnet already contains multicast group members, adding or removing a host to/from that group only requires VCs to be modified within that host's subnet. This is because the VCs external to that subnet are shared by other hosts. However if the new host is the only multicast group member in its subnet, in the Hop-by-Hop case, VCs between the local subnet router and the nearest router forwarding data for this multicast group must also be modified. In the Boundary Single Subnet approach, adding the first sender or receiver to the group also requires VCs to be created across the core network to the RP (if using a Shared Forwarding Tree), or to the edge of other subnets that already contain multicast group members (if using a Source Forwarding Tree).

In contrast, in the Cut-Through Single Subnet approach the entire network is treated as one subnet. This means that each time a host joins or leaves a multicast group, VCs spanning the entire network must be modified. This is regardless of whether the new receiver or sender is close to other hosts that are already members of this multicast group. Furthermore, the Hop-by-Hop approach uses a separate MARS and MCS within each subnet. This aims to keep multicast group management signalling latency low. In contrast the Cut-Through Single Subnet approach has to store the management information for the entire network in one entity. The Boundary Single Subnet approach is a hybrid between the two. Each subnet has a separate MARS which contains information about the hosts that are members of each multicast group. However there is a central MARS which maintains a list of the edge routers of each subnet that contains multicast group senders or receivers. The impact of the different multicast group management techniques employed by each inter-subnet approach on their signalling resource requirements is assessed in Section 8.4.4.

The performance of approaches are compared, again employing the JAMES network topology described in Section 6.2.3. Moreover the foreground traffic is assumed to follow the DIS traffic model and the background traffic the WWW model as detailed in Section 6.2.2. The performance of approaches are compared when adding senders or receivers to a multicast group comprising all ten DIS hosts shown in Figure 6.3. In this chapter, mean addition time results are presented. Delay percentiles were also calculated and the results were very similar. The reason why mean addition times are shown in this chapter is because most multicast applications do not have strict delay constraints on addition times, and thus mean performance is of more interest.
8.4 Performance Analysis

The time for either a new receiver to receive data from all senders, or, for data from a new sender to reach all receivers is of prime concern to users. The Cut-Through Single Subnet approaches create direct VCs across the ATM network. This chapter assumes that all new senders or receivers are located at the same network node. This may occur for instance if the node serves multiple users (e.g. a firewall on the edge of a network). However, in many environments, receivers and servers may be distributed further across the subnet. Hence this can be considered a worst case analysis. The impact of this assumption of approaches is discussed further below. As described above, the Hop-by-Hop and Boundary approaches perform differently when a new multicast host (sender or receiver) is the first host added to the multicast group in that subnet, than when some of the subnet’s hosts are already members of that multicast group. Both cases are considered.

8.4.1 Adding New Senders

Figure 8.3 shows the time for traffic from a new sender to reach all receivers as the sender addition rate varies. This includes (a) the time to obtain the receiver’s addresses from the MARS, (b) the time to create or modify any VCs required to allow the new sender to forward the traffic, and (c) the time for the new sender to forward traffic to the receivers (i.e. for the first application layer PDU to arrive at all receivers).

For the Hop-by-Hop approaches both best and worst cases are investigated. In the worst case the new sender joins a subnet that does not contain any other senders. This means that two sets of VCs need to be created: (a) within the subnet and (b) from the subnet router to a neighbouring router that already forwards traffic for that multicast group. In the best case, the subnet already contains senders for the same multicast group. This means that VCs only need to be modified or created within the subnet.

For the Boundary Single Subnet (SS) approaches best and worst cases are also considered. In the worst case when the sender joins a subnet where there are no other senders for that group, VCs need to be created: (a) within the subnet, and, (b) from the subnet router to either the RP (Shared Forwarding Tree case) or to all subnet routers that have receivers in their subnet (Source Forwarding Tree case).

In the Cut-Through Single Subnet (SS) approach, the new sender must create a pt-mpt VC to all receivers (or to the RP in the Shared Forwarding Tree case), regardless of whether the multicast group already contains other senders.
8. Multicast Dynamic Group Analysis

Figure 8.3: Mean Time for Data from a New Sender to reach all Receivers as the Sender Addition Rate Varies
Figure 8.3 compares the performance of the Cut-Through (SS), Boundary (SS) and Hop-by-Hop approaches when adding senders. Note Figure 8.3(b) does not include results for the best case Boundary Single Subnet Source Forwarding Tree approaches. This is because there is a negligible difference between the Source and Shared Forwarding Tree variants of the Boundary Single Subnet approach in the best case. This is due to the fact that VCs only need to be modified within the subnet. Hence the only difference is that the Source Forwarding Tree approach has slightly lower data packet forwarding delays.

Also note that Figure 8.3 only shows results for the Hop-by-Hop Shared Forwarding Tree approaches (not for the Hop-by-Hop Source Forwarding Tree approaches). This is because there is no difference between the approaches in the best case (other than the slightly higher forwarding delay of sending data traffic via the RP). In the worst case Hop-by-Hop approaches a difference only occurs if the new sender's subnet router is connected to several other routers. If a Shared Forwarding Tree is employed in this scenario, the subnet router would create a VC to its neighbouring router on the path to the RP. In the Source Forwarding Tree case the subnet router would need to create a pt-pt VC to each neighbouring router that represents the shortest path to one or more receivers. Even in this case there is only a significant difference between the Source and Shared Forwarding Tree variants of the Hop-by-Hop approach if there are a large number of next hop routers in the Source Forwarding Tree, and each of them is a large distance from the new sender's subnet router. Hence in the remainder of this chapter only the Hop-by-Hop Shared Forwarding Tree approaches are analysed.

As can be seen from Figure 8.3, regardless of the approach employed, the time to add a new sender is insensitive to the sender addition rate until over fifty senders are being added per second. This is because the time to forward data from the new sender to all receivers dominates the overall delay. That is, the time to modify the VCs is a small fraction of the time to forward the data from the new sender to all receivers. This analysis assumes a new sender transmits two 150 byte messages per second according to a Poisson distribution. Furthermore, the background data link and signalling link loads are assumed to be 0.7 and 0.4 respectively.

Figure 8.3 shows that the Cut-Through (SS), and Boundary (SS) worst case Source Forwarding Tree approaches produce the highest sender addition delays. In the Cut-Through (SS) approach, the new sender must create a pt-mpt VC to all receivers. In the Boundary (SS) worst case approaches the sender's subnet router must create a pt-mpt VC to all routers connected to subnets that contain receivers. The slight difference between the VC Mesh and MCS variants of the
Boundary (SS) approaches is due primarily to the higher data forwarding delay of the MCS approach as shown in Chapter 6. The signalling delay within the subnet is similar and the core network behaviour is independent of the intra-subnet approach employed. Figure 8.3 shows that these three multicast delivery approaches produce at least 50ms higher sender addition delays than all other approaches. Moreover these approaches are more sensitive to sender addition rate, only being able to support up to 50 new senders per second. All other approaches can support between 300 and 400 new senders per second.

The key point to note when comparing the Cut-Through (SS) and Boundary (SS) Source Forwarding Tree approaches is that the Cut-Through (SS) approach always has this performance. In the Boundary (SS) approach this addition delay is only incurred by the first sender to join a given subnet. If the subnet already contains other senders, the Boundary (SS) delay is around 100ms lower (as shown by the Boundary (SS) Shared Forwarding Tree best case results). Indeed the Boundary (SS) best case approach produces the lowest sender addition delay of all approaches. This is contrary to the common hypothesis in the literature that the Hop-by-Hop approach has the best performance when supporting dynamic multicast groups. The signalling delays of the Hop-by-Hop and Boundary (SS) approaches are identical in the best case since both approaches only need to modify VCs within the subnet. However as shown in Chapter 6, the Boundary (SS) approach produces lower data forwarding delays. Hence from the users perspective, the Boundary (SS) approach will add a new sender more quickly than the Hop-by-Hop approach.

The benefit of the Hop-by-Hop approach is that its worst case performance does not deteriorate to the extent of the Boundary (SS) approach with a delay increase of 10-15ms compared to a 40-45ms increase for the Boundary (SS) approach in the Shared Forwarding Tree case. In the Source Forwarding Tree case the difference between the Hop-by-Hop and Boundary (SS) worst case approaches is even greater. The Hop-by-Hop approach has a lower performance degradation than the Boundary (SS) approach because it only requires the subnet router to create a pt-pt VC to neighbouring routers that are on the route to the receivers (or to the neighbouring router on the route to the RP in the Shared Forwarding Tree case). In contrast the Boundary (SS) approach requires the subnet router to create a pt-mpt VC to all routers attached to subnets containing receivers (or a pt-pt VC to the RP in the Shared Forwarding Tree case). In most network topologies, a given router would not have a large number of neighbours, and these would be closer (in terms of distance and hops) than routers connected to the subnets containing receivers. Hence this analysis indicates that the Hop-by-Hop
8. Multicast Dynamic Group Analysis

approach has superior performance when adding a new sender to a subnet that does not contain any other senders.

The choice between the Boundary (SS) and the Hop-by-Hop approach will depend on whether senders tend to be distributed over many subnets or tend to be located in the same subnets, and hence the likelihood of a sender being added to a subnet that does not contain other senders. Figure 8.3 shows that there is a significant difference between the Hop-by-Hop and Boundary (SS) worst case approaches (particularly when the Source Forwarding Tree approach is employed), yet only a small difference in the best case scenario. Thus this suggests that the Hop-by-Hop approach should be employed rather than the Boundary (SS) approach. However before a firm recommendation can be made, the sensitivity of the approaches to the sender data message size and the performance when adding receivers needs to be considered.

Figure 8.3 shows that the Hop-by-Hop and Boundary (SS) worst case approaches can support 300 senders per second and the best case approaches 400 sender additions per second. The difference between the best and worst case variants of the Hop-by-Hop and Boundary (SS) approaches is the number of VCs the subnet switch and router at the edge of the subnet must support. In the best case only one VC needs to be created. However in the worst case two VCs need to be created through the subnet router, one within the subnet and one to either the neighbouring router (Hop-by-Hop case), or to the RP or other subnet routers (Boundary (SS) case). This means that although only 300 senders are being added per second, the signalling processors must create 600 VCs/s. This in addition to the background signalling traffic causes the signalling processors to overload.

Moreover, (contrary to commonly held hypotheses) the worst case Hop-by-Hop and Boundary (SS) Shared Forwarding Tree approaches over-utilise the signalling network at a lower sender addition rate (300 senders/s) than the Cut-Through (SS) Shared Forwarding Tree approach (400 senders/s). In the Hop-by-Hop and Boundary (SS) Shared Forwarding Tree worst case approaches new VCs must be created: (1) within the subnet and (2) between the subnet router and its closest neighbour that already receives traffic for this multicast group. This places a high signalling load on both the subnet ATM switch and the subnet router. In contrast, in the Cut-Through (SS), Shared Forwarding Tree approach, each new sender only requires one VC to be created. Hence the Cut-Through (SS) Shared Forwarding Tree approach is more robust to sender addition rate than the Hop-by-Hop and Boundary (SS) worst case approaches.

In an operating scenario where all new senders are located in the same subnet, the Cut-Through (SS) Shared Forwarding Tree, Boundary (SS) and Hop-by-Hop
best case approaches have the same sensitivity to sender addition rate. In all of these cases only one VC needs to be created. However, we hypothesise that if new senders are distributed across multiple subnets the Boundary (SS) and Hop-by-Hop approaches will place lower demands on the signalling network than the Cut-Through (SS) Shared Forwarding Tree approach. This hypothesis is examined in Section 8.4.4

Figure 8.3 shows that the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches take longer to add senders compared to their equivalent Shared Forwarding Tree approaches. If Source Forwarding Tree approaches are employed VCs must be created across the entire network. In contrast the Shared Forwarding Tree approach only needs to create one pt-pt VC to the RP. Thus even though it takes longer to forward traffic via a RP, the difference in signalling delay causes the Shared Forwarding Tree approach to add new senders faster than the Source Forwarding Tree approach. In the intra-subnet multicast approaches we found the reverse was true with the VC Mesh approach, out-performing the MCS approach (even though the MCS has lower signalling delays). This is because these approaches only need to create VCs over a relatively short distance, which means the extra signalling penalty is small. However, it is also important to note that even in the best case we assume there is only one other group member in the subnet.

If there were many receivers in the same subnet we hypothesise the difference in the signalling delay between the VC Mesh and MCS approaches would increase (because the VC Mesh approach must create a pt-mpt VC to all local receivers), potentially to the extent that the MCS approach would out-perform the VC Mesh approach in terms of overall sender addition delay.

We investigated the impact of the number of receivers on the VC Mesh and MCS approaches, assuming the subnet already contains senders. The analysis indicated that there is no significant difference between the approaches (on the order of a few milliseconds) even when the subnet contains seven receivers, when one sender is added per second. A significant difference between the approaches only occurs when the signalling network is highly utilised. In our analysis this only occurred when the sender addition rate was on the order of hundreds per second, or when the subnet contained hundreds of receivers.

The impact of the number of senders already present in the subnet on the MCS and VC Mesh approaches was also considered. This indicated that when the subnet contained no senders, the MCS approach takes longer to add the first sender. However the time to add subsequent senders is similar for both approaches. Hence our analysis indicates that there is no significant difference between the VC
Mesh and MCS approaches when the signalling network is lightly to moderately utilised, with the VC Mesh approach producing 5-10 ms lower overall addition delays. Only when the signalling network is extremely highly utilised does the MCS approach significantly out-perform the VC Mesh approach.

Our analysis also indicated that in geographically large subnets the MCS approach will also be superior, if the MCS is located in the centre of the network. This is because the propagation delay (and cumulative processing delay at intermediate switches) will be lower because the pt-pt VC only needs to be created half-way across the network. If VCs are created directly between senders and receivers, signalling traffic must traverse the entire subnet.

This analysis has shown that employing the Hop-by-Hop approach means that traffic from a new sender will reach receivers much faster than if the Cut-Through Single Subnet approach is followed. Moreover, the Hop-by-Hop approaches can support a higher number of new senders. The Boundary Single Subnet approach out-performs the Hop-by-Hop approach in the best case scenario where the subnet contains other senders. However, when adding the first sender to a subnet, the Hop-by-Hop provides lower sender addition delays and less demand on the signalling network. Moreover, in the worst case scenario there is a significant difference between the two approaches, whereas the delay difference is small in the best case. Therefore this analysis shows that the Hop-by-Hop approach provides superior dynamic group support, both in terms of delay and signalling network requirements.

8.4.2 Adding New Receivers

Figure 8.4 compares the time for data to reach a new receiver when each multicast delivery approach is applied. One can see that the relative performance of approaches is the same as when a new sender is added. The delay calculation includes: (1) the time to modify VCs to add the receiver and (2) the time for the first data packet (from any of the senders) to reach the receiver.

As was the case when adding a new sender, all approaches are insensitive to receiver addition rate until over fifty new receivers are added each second. Again the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches are the first to over-utilise the signalling network. This is because in the Cut-Through case each receiver addition triggers all senders to add that receiver to their pt-mpt VC. Similarly in the Boundary (SS) case all subnet routers connecting to senders must add the receiver’s subnet router to their pt-mpt VC.

Figure 8.4 also shows that there is a greater difference (than when adding senders) between where the worst case Hop-by-Hop and Boundary (SS) Shared
8. Multicast Dynamic Group Analysis

Figure 8.4: Mean Time for Data to reach a New Receiver as the Receiver Addition Rate Varies
Multicast Dynamic Group Analysis

Forwarding Tree approaches over-utilise the signalling network (300 receivers/s) compared to the Cut-Through (SS) Shared Forwarding Tree approach (500 receivers/s). This is because now the Cut-Through (SS) Shared Forwarding Tree approach only requires one new leaf to be added for each new receiver, compared to a new VC for each sender. The volume of signalling traffic required to add a leaf is lower and hence the signalling network can handle a higher receiver addition rate than sender addition rate.

In Figure 8.4 there is a smaller delay difference between the Cut-Through (SS) Source and Shared Forwarding Tree approaches compared to Figure 8.3. Similarly for the Boundary (SS) worst case Source and Shared Forwarding Tree approaches. This is because a new pt-pt VC must be created before leaves can be added to it when adding a sender via a Source Forwarding Tree. However, when adding a sender to a Shared Forwarding Tree only a pt-pt VC needs to be created. In contrast, when adding receivers, both Source and Shared Forwarding Trees only require new leaves to be added to existing pt-mpt VCs. Since multiple leaves can be added in parallel, the signalling delay difference between the Forwarding Tree approaches when adding receivers is due only to the extra distances to be covered and additional queuing delays.

Another factor that must be considered is the scalability of approaches, particularly from the operators viewpoint. Both Figures 8.3 and 8.4 have shown that the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches can support less than one hundred new receivers or senders per second. Employing the Shared Forwarding Tree approach rather than the Source Forwarding Tree approach alleviates this problem because it only modifies one VC between the RP and the new host. However, all of the signalling links and processors around the RP will be heavily loaded regardless of how the hosts are distributed across the network. The only time the signalling load would be better distributed is if UNI 4.0 pt-mpt leaf prompted join without root notification is employed. In this case only signalling processors on the new branch of the pt-mpt VC need to be involved. In contrast, the Boundary (SS) best case and Hop-by-Hop approaches are usually more scalable, particularly if new hosts are distributed rather than localised in one subnet (regardless of the ATM signalling version employed). This is because these approaches only require signalling around the edge of the network. Thus the operator's core network signalling processors and links remain lightly loaded. The signalling network requirements of each approach are investigated further in the next section.

Like the case when adding senders, there is a negligible difference between the Hop-by-Hop and Boundary (SS) best case approaches when adding receivers, but
8. Multicast Dynamic Group Analysis

A greater difference (on the order of 15ms) in the worst case. Moreover we hypothesise that the difference between the approaches in the worst case will be even greater if new receivers are distributed around multiple subnets. Hence the Hop-by-Hop approach seems to be the preferred approach when adding either senders or receivers to a multicast group. The only situation where the Cut-Through (SS) approaches may be superior is if large messages are being transmitted and the delay forwarding benefits outweigh the signalling delay penalties of the Cut-Through (SS) approaches. We analyse the sensitivity of approaches to sender message size in the next section. However it is important to note here that if the Cut-Through (SS) approach is employed it should be in conjunction with a Shared Forwarding Tree not with Source Forwarding Trees. This has little impact on delay but a significant impact on the demands placed upon the signalling network.

8.4.3 Sensitivity of Addition Delay to Data Message Size

In the previous sections, the analysis has assumed that the data messages are 150 bytes in length. This section analyses the sensitivity of addition delay to the size of the data message being forwarded.

Figure 8.5 shows the time for data from a new sender to reach receivers as the size of its data message varies, assuming one new sender is added per second. Similar results were obtained when adding new receivers. This shows that if the sender message size is smaller than 5 kbytes, then the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches have the highest sender addition delay as discussed in Section 8.4.1. However, if the sender message size is larger than this, the Cut-Through (SS) Source Forwarding Tree approach starts to out-perform the Hop-by-Hop and Boundary (SS) approaches. Indeed, if the message size is greater than 20 kbytes, only the Boundary (SS) VC Mesh best case approach has a lower sender addition delay. Once the message size exceeds 50 kbytes, the Cut-Through (SS) Source Forwarding Tree approach has the best performance. Similar results were obtained when adding new receivers.

The reason why all other approaches are more sensitive to sender message size than the Cut-Through (SS) Source Forwarding Tree approach is that they all require reassembly and IP level processing at either routers, MCSs or RPs. The reassembly delay is directly related to the size of the message and the volume of messages in the network. Figure 8.5 also shows that the Boundary (SS) approaches are less sensitive to increasing message size than the Hop-by-Hop approaches. This is also because the Boundary (SS) approaches require less reassembly across the network than the Hop-by-Hop approaches.

To summarise, Figure 8.5 has shown that the Cut-Through (SS) approaches
Figure 8.5: Time for Data from a New Sender to reach Receivers as the Sender's Data Message Size Varies
can out-perform the Hop-by-Hop and Boundary (SS) approaches, when large message sizes are involved. This is due to their superior forwarding delay performance (see Chapter 6). Therefore the Cut-Through Single Subnet, Shared Forwarding Tree approach may be applicable for real-time multicast traffic with strict delay requirements when carrying moderate to large messages. However the impact on the signalling network must also be considered. The analysis in this chapter has indicated that the Cut-Through Single Subnet Source Forwarding Tree approach places higher demands on the signalling network than other approaches. It is also sensitive to the number of multicast group members. The signalling demands of approaches are considered further in the next section.

8.4.4 Signalling Demands of Approaches

Network operators are also concerned about the resources required to provide dynamic multicast group support. The previous portion of this chapter has focused on the relative delay performance of approaches. In addition some indication of the signalling demand of approaches is given by analysing how many new hosts (receivers or senders) each approach can add per second.

<table>
<thead>
<tr>
<th>Approach</th>
<th>Number of New Receivers Added Per Second</th>
<th>Number of New Senders Added Per Second</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-Through (SS), Source</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Cut-Through (SS), Shared</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>Boundary (SS), MCS, Source, Worst Case</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>Boundary (SS), VC Mesh, Source, Worst Case</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Boundary (SS), MCS, Shared, Worst Case</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Boundary (SS), VC Mesh, Shared, Worst Case</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>Boundary (SS), MCS, Shared, Best Case</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>Boundary (SS), VC Mesh, Shared, Best Case</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>Hop-by-Hop, MCS, Shared, Worst Case</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Hop-by-Hop, VC Mesh, Shared, Worst Case</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>Hop-by-Hop, MCS, Shared, Best Case</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>Hop-by-Hop, VC Mesh, Shared, Best Case</td>
<td>500</td>
<td>400</td>
</tr>
</tbody>
</table>

Table 8.2: Number of New Receivers and Senders each Approach can Support

Table 8.2 shows how many new senders and receivers each approach can support before the signalling network overloads given the assumptions of Sections 8.4.1 and 8.4.2. This table highlights two areas: Firstly the Boundary (SS) worst case and the Cut-Through (SS) Source Forwarding Tree approaches have significantly greater signalling resource requirements than the other approaches. Secondly,
adding a new sender is in general more demanding than adding a new receiver.

The distribution of signalling processor capacity requirements when adding senders is examined in Figure 8.6. Figures 8.6(a) and (b) assume that the new senders are (a) located in one subnet or (b) uniformly distributed across the network respectively. This analysis assumes that: 100 senders are being added per second, and the JAMES network topology from Chapter 6 is employed, hence there are 36 ATM switches within the network in total. It may be unlikely for a single multicast group to add 100 senders per second. However in the backbone, senders will be being added to many different groups. In this environment it is entirely possible that the aggregate sender addition rate is on the order of 100 senders/s. It is assumed that on portions of a link where a pt-mpt VC does not already exist it takes the same resources to add a new leaf as to create a pt-pt VC. However, on portions of the link where the pt-mpt VC already exists we assume that only 20% of the resources necessary to create a new pt-pt VC are needed to add a leaf.

Figure 8.6(a) indicates that for all approaches except the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches, over thirty of the switches are not processing any signalling traffic. Thus, the addition of a new sender only affects a small number of switches. In contrast, in the Cut-Through (SS) Source Forwarding Tree approach twenty-five of the switches are processing signalling traffic. This is because this approach requires each new sender to create a pt-mpt VC to all multicast group receivers. Similarly, in the Boundary (SS) worst case Source Forwarding Tree approach, fifteen switches are processing signalling traffic. The number of switches involved is lower than in the Cut-Through (SS) case because the VC changes only extend to the subnet routers connected to receivers, rather than to the receivers themselves. This means that the internal subnet switches impacted in the Cut-Through (SS) case are not involved in the Boundary (SS) case, even in the worst case scenario. This highlights the major benefit of the Boundary (SS) approach compared to the Cut-Through (SS) approach of aggregating traffic at the edge of the core network.

The histogram shows the disadvantage of the Cut-Through (SS) and to a lesser extent the Boundary (SS) worst case approaches in terms of the demands placed on the signalling network when adding senders. All other approaches require either only one VC to be created at the Shared Forwarding Tree RP, or the scope of the pt-mpt VC is restricted to within the new sender's subnet (or in the Hop-by-Hop worst case the VC extends to the neighbouring router). Hence from a signalling

---

1 The Hop-by-Hop and Cut-Through approaches are indicated by solid lines and the Boundary approaches by dotted lines.
Figure 8.6: Signalling Processor Requirements When Adding Senders
network perspective the Cut-Through Single Subnet Source Forwarding Tree approach is not recommended. Moreover the Boundary (SS) Source Forwarding Tree approach should not be employed if most new senders join subnets that do not already contain senders.

In all approaches some signalling processors require a capacity of 100 VCs/s. This is at the limit of the capabilities of most currently deployed backbone ATM switches. Indeed even recently announced backbone ATM switches can only create 300 VCs/s [Tec98]. In the Cut-Through Single Subnet Source Forwarding Tree approach, nineteen of the thirty-six signalling processors require a capacity of 100 VCs/s. Six other switches require between 100 and 300 VCs to be created per second. This is assuming a network spanning Europe. In inter-continental networks future aggregate sender addition rates may be even greater than the rate of 100 senders/s assumed in this analysis. This clearly shows that if multicast applications are to be widely deployed over ATM infrastructure, the signalling capabilities of ATM switches need to be improved, or approaches like the Hop-by-Hop approach which only impact switches at the edge of the network must be employed.

The Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches are so demanding because the new pt-mpt VC must span the core network. The next worst approaches are the Cut-Through (SS) and Boundary (SS) worst case Shared Forwarding Tree approaches. These approaches have to create a new pt-pt VC from the sender to the RP. Given the RP is often located at the wire-line centre of the network, in large networks this can involve even greater numbers of ATM switches than found in the JAMES network. In the Cut-Through (SS) case this requires four switch signalling processors to process the equivalent of 100 new VCs/s, because the VC is created directly from the sender to the RP. The situation is worse if the Boundary (SS) approach is employed because VCs must be created within the subnet, and then another VC from the subnet router to the RP. Hence the subnet router must process two new VCs for each new sender. In the MCS case the subnet switch also has to process two new VCs: one between the sender and MCS, and the other from the MCS to the router. This highlights the disadvantage of terminating VCs at the edge of the core network.

The Hop-by-Hop approaches involve a smaller number of signalling processors when adding a sender, compared to the Cut-Through (SS) and Boundary (SS) approaches. However the signalling load on the subnet switch and router is the same as the Boundary (SS) Shared Forwarding Tree approach in the worst case scenario. In the Hop-by-Hop and Boundary (SS) best-case scenarios only the subnet switch and edge router have to process signalling traffic (indeed if the
MCS approach is employed, the edge router does not need to process the VC either). Hence these approaches place much lower demands on the signalling network especially if the new senders are distributed across multiple subnets as shown in Figure 8.6(b).

From this analysis it is clear that the Cut-Through (SS) Source Forwarding Tree approach places the heaviest demands on the signalling network. The Boundary (SS) worst case Source Forwarding Tree approaches also place heavy demands on the signalling network. In all other approaches far fewer switches/routers need to process signalling traffic when adding a new sender. Moreover those that do, process at most 200 VCs/s when adding 100 senders/s, whereas the three Source Forwarding Tree approaches can require some switches to process up to 280 VCs/s.

As discussed in the previous sections, Figure 8.6(a) assumes that all new senders are located in one subnet. In practice new senders are likely to be distributed across the entire network. This will increase the number of switches processing signalling traffic, but will mean that rather than all of the signalling traffic being localised at one switch, it will be distributed across multiple switches. In this case each individual switch will be utilised far more lightly.

Figure 8.6(b) compares the signalling processor requirements of approaches when senders are evenly distributed across all eighteen subnets comprising the JAMES network. This shows that both the Cut-Through Single Subnet Source and Shared Forwarding Tree approaches still have signalling processors with moderate to high capacity requirements. In the Source Forwarding Tree approach the number of signalling processors with high capacity requirements (demand is over 100 VC setups/s) remains similar with 8, 13, 1 and 1 processors with capacities of 100, 120, 140 and 220 VCs/s respectively; compared to 19, 2, 1, and 3 processors with capacities of 100, 120, 220, and 280 VCs/s when all new senders are in the same location. In contrast in the Cut-Through Single Subnet Shared Forwarding Tree approach, four processors originally required a capacity of 100 VC setups/s, compared to one processor with a capacity of 100 VCs/s when senders are distributed. This is because the central RP must still process signalling traffic for all new senders, regardless of their location. However the signalling traffic will follow different routes to the RP, compared to the centralised case where only one route is followed (see Figure 6.1 for the network topology). Hence the other switches in the network have to process far fewer VCs.

In the Boundary (SS) worst case approaches, the difference in performance between centralised and distributed sender locations is similar to the Cut-Through (SS) cases. For instance the Boundary (SS) Source Forwarding Tree VC Mesh worst case signalling requirements change from 12, 1, and 2 at 100 VCs/s, 220
VCs/s and 260 VCs/s to 12, 1, and 1 at 100 VCs/s, 140 VCs/s and 220 VCs/s. In the Boundary (SS) VC Mesh Shared Forwarding Tree worst case signalling requirements change from 3 at 100 VCs/s and 1 at 200 VCs/s to 1 at 100 VCs/s. This highlights that for both Cut-Through (SS) and Boundary (SS) worst case approaches, the Source Forwarding Tree approaches are relatively insensitive to the location of senders. However, when employed with Shared Forwarding Trees the signalling network demands are much lower, particularly if senders are distributed over many subnets.

In Figure 8.6(a), all Hop-by-Hop and Boundary (SS) best case approaches had at least one signalling processor with a capacity requirement of 100 VC setups/s. Figure 8.6(b) clearly shows the benefits of the Hop-by-Hop approaches compared to the Cut-Through (SS) approaches when new senders are distributed across multiple subnets. Even in the Hop-by-Hop worst case approaches, no signalling processor needs a capacity over 30 VC setups/s. This is because at worst only the subnet switches and the neighbouring switch are involved in the new sender addition process. Of the Hop-by-Hop approaches, the Hop-by-Hop MCS best case approach has the lowest signalling requirements with each local subnet switch adding one leaf for each new sender. The subnet router does not need to process any signalling traffic. In contrast in the Hop-by-Hop VC Mesh best case approach, both the local switch and subnet router must process traffic to create the pt-mpt VC.

In the worst case the relative performance of the intra-subnet approaches is reversed. This is because the MCS approach requires three VCs to be created, whereas the VC Mesh approach requires only two VCs as discussed above. However, as shown in Figure 8.6 there is no significant difference in the signalling network demands of the VC Mesh and MCS best case approaches, particularly when new senders are distributed across the entire network. The Boundary (SS) approaches have the same signalling requirements as the Hop-by-Hop approaches in the best case. However, Figure 8.6 shows that in the worst case, the Boundary (SS) approach has significantly higher signalling requirements, particularly if employed in conjunction with a Source Forwarding Tree.

To summarise, the analysis where new senders are distributed across the network has shown that the Hop-by-Hop approaches are superior to the Cut-Through Single Subnet and Boundary Single Subnet worst case approaches in terms of signalling network requirements. Furthermore, there is little difference in the signalling network performance of the Hop-by-Hop MCS and VC Mesh approaches when considering small subnets with few receivers. However the MCS approach will out-perform the VC Mesh approach when the subnet contains a large num-
number of multicast group receivers, to which the new sender must connect. Hence characterisation of multicast group member locations is important when selecting intra-subnet delivery approaches. Figure 8.6 has also highlighted the impact the sender location distribution can have on the performance of the multicast delivery approaches, particularly on approaches that employ Shared Forwarding Trees, or the Hop-by-Hop inter-subnet approach.

We also considered the signalling processor capacity requirements when adding new receivers. Figure 8.7 shows the signalling processor requirements of approaches when adding 100 new receivers per second to multicast groups. All new receivers are assumed to be located in the same subnet. As in the case of adding senders, only the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches utilise a large number of signalling processors. In this case however, most of the signalling processor require capacities of 20 VC setups/s rather than 100 VC setups/s. This is because adding receivers to existing pt-mpt VCs means only a small portion of the pt-mpt path needs to be created. As discussed above, it is assumed that on portions of a link where a pt-mpt VC does not already exist it takes the same resources to add a new leaf as to create
8. Multicast Dynamic Group Analysis

a pt-pt VC. However, on portions of the link where the pt-mpt VC already exists we assume that only 20% of the resources necessary to create a new pt-pt VC are needed to add a leaf. Hence in general, less demand is placed on most signalling processors when adding receivers.

Figure 8.7 also shows that some signalling processors are extremely highly utilised when adding receivers. For instance two switches require capacities of 1000 VC setups/s, which is much greater than the capacity of the newest backbone ATM switches (in contrast when adding a new sender the greatest demand was 280 VCs/s). The switches with these high signalling demands are those close to the new receiver. These switches must create a new leaf for every sender in the multicast group if Source Forwarding Tree approaches are employed since the pt-mpt VC does not already have a branch at these switches. In contrast when adding a sender, the leaves are being added to an existing VC at switches close to the sender, hence the signalling demands are much lower. This again highlights the unsuitability of the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approach when supporting dynamic multicast groups.

To conclude, the signalling processor requirements analysis has shown that the Cut-Through (SS) Source Forwarding Tree approach should not be employed when supporting dynamic multicast groups, since it places significantly higher demands on the signalling network than all other approaches. Furthermore, the Boundary (SS) Source Forwarding Tree approach has similar performance to the Cut-Through (SS) Source Forwarding Tree approach in the worst case. Hence in scenarios where senders or receivers are being added to many different subnets, the Boundary (SS) Source Forwarding Tree approach should also be avoided.

This analysis has also shown that the Hop-by-Hop approaches require much lower signalling processor capabilities than the Cut-Through (SS) and Boundary (SS) worst case approaches, particularly when new senders or receivers are distributed across multiple subnets. Hence in terms of dynamic multicast group support, the Hop-by-Hop approach is the best approach. In environments where new hosts are being added to subnets that already contain multicast group members, the Boundary (SS) approaches provide slightly lower addition delays. However, if hosts are being added to subnets with no other multicast group members, the Hop-by-Hop approaches produce significantly better performance both in terms of addition delay and signalling resource requirements.
8.5 Conclusions

This chapter has investigated the problem of dynamic multicast group support. Twelve alternative multicast delivery approaches have been compared in terms of their delay performance and signalling network resource requirements. Our findings are listed below, in terms of the objectives detailed at the start of the chapter.

- Determine if there is a significant difference in the time it takes the MCS and VC Mesh approaches to add (or remove) senders or receivers.

Contrary to commonly held hypotheses our analysis showed that even when the subnet contains many receivers and a new sender is being added each second, there is no significant difference in the signalling network delays of the two approaches. The same held true when adding a receiver to a subnet containing many senders. Significant differences in signalling network performance only occurred: when adding hundreds of new senders or receivers a second, when the subnet covers large distances, or when the signalling network is already highly utilised. This means that the VC Mesh approach typically outperforms the MCS approach when considering overall addition delay. This is because it has lower forwarding delays (between five and ten milliseconds lower) since traffic does not need to travel via a MCS. However, the MCS typically places lower demands on the signalling network than the VC Mesh approach if the subnet already contains other multicast group members.

- Investigate whether the time to add new senders or receivers significantly differs between the alternative inter-subnet multicast delivery approaches.

The Cut-Through (SS) and Boundary (SS) worst case approaches can take significantly longer to add a new sender or receiver compared to the Hop-by-Hop approach, particularly when employed in conjunction with a Source Forwarding Tree. This is the case even when the multicast group contains only ten senders/receivers. When multicast groups contain more senders/receivers this difference will become even more significant. This is because the Hop-by-Hop approach localises changes, whereas adding a new sender or receiver in the Cut-Through (SS) or Boundary (SS) worst case approach can impact the entire network. Thus, even though the Cut-Through (SS) and Boundary (SS) approaches have lower forwarding delays (see Chapter 6) than the Hop-by-Hop approach this is outweighed by the time to modify the VCs to
add the new sender or receiver. The only circumstances we found where this did not hold is if the new sender forwards large packets.

Moreover in general the Cut-Through (SS) and Boundary (SS) worst case approaches are less scalable than the Hop-by-Hop approaches, regardless of how the senders and receivers are distributed across the network. This is because the addition of new hosts causes VCs to be modified across the core network (to RPs in the case of shared Forwarding Trees, or to hosts, or host subnet routers in the case of Source Forwarding Trees). In contrast the Hop-by-Hop approaches only impact switches and routers around the edge of the network. Thus, particularly when new hosts (senders or receivers) are distributed across many subnets, there is a significant difference in the demands placed on the signalling network, and hence also the time to create/or modify the VCs to add the new host.

One surprising observation from this chapter however is that the Cut-Through (SS) Shared Forwarding Tree approach can support more sender or receiver additions than the Hop-by-Hop and Boundary (SS) worst case approaches, if all new hosts are located in the same subnet. This is because, although the Cut-Through (SS) Shared Forwarding Tree approach impacts a higher number of switches (all those on the path to the RP), each of them only needs to modify one VC per sender. In contrast in the worst case Hop-by-Hop and Boundary (SS) approaches, the subnet router needs to process two VCs per new host, one within the subnet and one within the core network. However, this chapter did show that if new hosts are distributed across several subnets, the Hop-by-Hop approach is more scalable than the Cut-Through (SS) Shared Forwarding Tree approach, if the new hosts are distributed over multiple subnets.

In the best case scenario where the new host joins a subnet that already contains other multicast group members, the Boundary (SS) approach provides the lowest addition delay. However the difference between the Hop-by-Hop and Boundary (SS) best case addition delays is negligible and their demands on the signalling network are identical. As described above, in the worst case scenario where the new host is the first to join a subnet, the Hop-by-Hop delay and signalling performance is significantly better than the Boundary (SS) approach, particularly if the approaches are employed with Source Forwarding Trees. Hence we recommend that the Hop-by-Hop approach be employed when supporting dynamic multicast groups. The Boundary approach should only be employed if the majority of new hosts join subnets
that already contain other multicast group members. The only circumstance in which the Cut-Through (SS) Shared Forwarding Tree approach should be employed is if the senders are sending messages that are several kilobytes in length, and the signalling network has sufficient resources to handle the additional signalling traffic.

- Examine the difference between Source and Shared Forwarding Trees when supporting dynamic multicast groups.

This chapter has shown that significant benefits can be gained by employing the Shared Forwarding Tree approach rather than the Source Forwarding Tree approach, both in terms of delay and signalling resource requirements. This is when the Forwarding Tree approach is employed in conjunction with either the Cut-Through (SS) or Boundary (SS) approach. There is not a significant difference when employed with the Hop-by-Hop approach because the Hop-by-Hop approach only modifies VCs around the edge of the network, regardless of whether a Source or Shared Forwarding Tree approach is employed.

The difference between the Forwarding Tree approaches is particularly significant when adding senders. This is because the Source Forwarding Tree approach requires an entire tree to be created, whereas the Shared Forwarding tree approach requires only one pt-pt VC to be created, regardless of the size of the multicast group. Moreover, the Shared Forwarding Tree approach is significantly more scalable in terms of the number of new senders it can add per second, particularly if the multicast group contains many receivers. For example, in our analysis the Source Forwarding Tree approach could only support 50 new senders/s, whereas the Shared Forwarding Tree approach could support 400 new senders/s when the multicast group only contained ten receivers.

When adding receivers the delay difference between Source and Shared Forwarding Trees is smaller, (on the order of five milliseconds). This is because both approaches only require leaves to be added to existing VCs. Although the Source Forwarding Tree approach requires multiple leaves to be added these can all be added in parallel. Hence when adding receivers there is not a significant delay difference between the Cut-Through (SS) Source and Shared Forwarding Tree approaches. However, there is a significant difference (an order of magnitude) in the signalling resource requirements of the approaches (even greater than when adding senders). Hence, regardless of whether adding senders or receivers, the Shared Forwarding Tree approach
provides significantly better performance than the Source Forwarding Tree approach.

- Determine the relative signalling network demands of the intra-subnet, inter-subnet and alternative forwarding tree approaches.

Comparing the signalling processor requirements of all approaches it is clear that the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approaches have significantly higher signalling network demands than all other approaches. When adding receivers particularly, the Cut-Through (SS) and Boundary (SS) worst case Source Forwarding Tree approach can require signalling network capacities far in excess of the capabilities of existing ATM switches. Moreover, this highlights that the hypothesis that the traditional ATM Single Subnet performance is not sufficient to adequately support dynamic multicast groups is correct. Hence there is a need for further study to compare the alternative LSR and NHRP Router approaches.

The signalling processor requirements of all other approaches were similar if the senders or receivers all join the same subnet. If the new senders or receivers are distributed across the network, the Hop-by-Hop (and Boundary (SS) best case) approaches place significantly lower signalling demands on the network than the Cut-Through (SS) (and Boundary (SS) worst case) Shared Forwarding Tree approaches. The analysis also indicated that in the majority of network environments (particularly if new senders or receivers are well distributed) there is no significant difference between the signalling requirements of the VC Mesh and MCS approaches.

- Determine which of the multicast delivery component choices has the greatest impact on the dynamic group support performance of the multicast delivery systems.

The choice of inter-subnet approach had the greatest impact on performance. This is especially true when the location of senders or receivers are distributed across the entire network. However the choice of Forwarding Tree approach can also significantly impact performance, particularly when employed in conjunction with either the Cut-Through (SS) or Boundary (SS) inter-subnet approach.

To summarise, this analysis has clearly shown that regardless of whether adding new senders or receivers, the Cut-Through, Single Subnet, Source Forwarding Tree approach should not be employed for dynamic multicast groups.
This is because of the large addition delays, the sensitivity of this approach to addition rate, and the heavy demands it places on the signalling network. The only circumstance where the Cut-Through (SS) approach may be considered is when senders are forwarding large messages (on the order of tens of kibytes) and the application is delay sensitive. In this situation it provides better addition delay performance than the other approaches. In general the findings of this chapter support the hypothesis that the traditional ATM Single Subnet performance is not sufficient to adequately support dynamic multicast groups. Hence there is a need for further study to compare the alternative LSR and NHRP Router approaches.

This chapter has also shown that even though the Boundary (SS) approach potentially offers the lowest addition delays, it should only be employed if the majority of new hosts (senders or receivers) join subnets that already contain multicast group members. If this is not the case and the host is the first group member in a subnet, The Boundary (SS) approach performance deteriorates significantly. Moreover we found that when the subnet already contains multicast group members the delay difference between the Hop-by-Hop and Boundary (SS) approaches is insignificant. However, when joining a host to a new subnet, the Hop-by-Hop approach performance is superior to the Boundary Single Subnet approach.

We therefore conclude that the Hop-by-Hop Shared Forwarding Tree approach should be employed for supporting dynamic multicast groups. This is because, both the signalling network resource requirements and delays are significantly lower than for the Cut-Through Single Subnet and Boundary Single Subnet worst case approaches. Furthermore, we found there is no significant difference between the Hop-by-Hop VC Mesh and MCS approaches unless the signalling network is very highly utilised.

Our analysis also indicated that there is a significant difference between the Source and Shared Forwarding Tree approaches when employed in conjunction with either the Boundary (SS) worst case or Cut-Through (SS) approaches. Hence if an operator does decide to employ either the Boundary (SS) or Cut-Through (SS) approaches is should be in conjunction with the Shared Forwarding Tree approach. This provides significant savings both in terms of addition delay and signalling resource requirements.
Chapter 9

Conclusions

9.1 Overview

Many schemes for providing Internet services over ATM have been proposed. However, a detailed review of the literature reveals that little analysis comparing these schemes has been performed, particularly when considering wide area ATM networks. A review of the literature also reveals that the selection of a delivery mechanism is central to the overall problem of providing Internet services over ATM. Hence this thesis has provided a detailed quantitative analysis of approaches for providing unicast and multicast IP traffic delivery over wide area ATM networks.

This chapter highlights the major findings of this thesis and summarises our recommendations to network operators when selecting an IP over ATM delivery mechanism. Areas for further study are also presented.

9.2 Major Findings

9.2.1 Unicast Delivery

- The Cut-Through unicast delivery approach provides significant response time savings over the Hop-by-Hop approach, even if the direct VC must be created, as long as the signalling network can support the expected volume of signalling traffic.

- The Hop-by-Hop approach has significantly lower resource requirements than the Cut-Through approach. However, contrary to commonly held hypotheses, the VC and signalling resource requirements of the Cut-Through approach can easily be supported by current ATM equipment. Given current switches can process around 100 VC setups/s a single switch can support
the traffic generated by 3000 HTTP proxies.

- Our analysis has shown that it is better to buffer traffic while a Cut-Through VC is being created, rather than forwarding the traffic via a Hop-by-Hop VC until the Cut-Through VC exists. This is in contrast to the hypothesis commonly held within the literature that the Hybrid approach out-performs the Buffered approach.

- In circumstances where the signalling network is highly utilised the Hop-by-Hop approach should be employed if the Cut-Through VC does not exist. This is because the Buffered approach produces high signalling network delays. Furthermore, the Hybrid approach wastes signalling resources since all of the traffic reaches its destination via the Hop-by-Hop VCs before the Cut-Through VC is open.

- Our analysis has shown that the selection of the optimal VC holding time is critical for the Cut-Through approach. Furthermore it must be selected on the basis of the relative costs of maintaining a VC and creating a new VC. When these factors are equally waited we found the optimal VC holding time is 5 minutes. However if the relative importance of these factors varies between 0.1, and 10, the optimal holding time varies between 30 seconds and twenty minutes. In contrast, when the Hop-by-Hop approach is employed, it is best to keep VCs open semi-permanently.

- The analysis has shown that in most circumstances there is not a significant delay penalty associated with creating VCs for the Cut-Through approach. Moreover, the resource requirements of the Cut-Through approach can be met by current ATM equipment in most network scenarios. Situations where the VC creation delay penalty can be high is if the link utilisation is low, the volume of traffic being transmitted is low, or the signalling network is highly utilised. Therefore there may be network environments where Label Switching approaches can provide better performance than the Cut-Through approach, if it can create an end-end direct VC more quickly. This is an area for further study.

9.2.2 Multicast Delivery

Forwarding Delay

- The MCS and Shared Forwarding Tree approaches are extremely sensitive to the distance, number of hops, and number of interfaces to the MCS and
9. Conclusions

RP respectively. The closer the RP or MCS to the wire-line centre of the network or subnet, the better the performance. Furthermore the analysis shows that the smaller the number of interfaces to the MCS or RP the higher the impact on response time performance compared to the VC Mesh and Source Forwarding Tree approaches.

- When the links to RPs and MCSs are engineered adequately, there is no significant difference between the VC Mesh and MCS intra-subnet approaches, or between Source and Shared Forwarding Trees. Hence the choice of inter-subnet delivery approach has the greatest impact on overall multicast forwarding delay.

- In most operating network environments the Hop-by-Hop approach produces significantly higher delays than the Boundary and Cut-Through approaches. This is particularly true when the network is heavily utilised or when there are many hops separating the RP from the senders and receivers. Moreover in the majority of network environments there is no significant performance difference between the Cut-Through and Boundary VC Mesh approaches, unless the data network is heavily utilised.

- The jitter in the response time observed by different receivers is significant in heavily utilised networks, representing half the magnitude of response time.

- No multicast delivery system is able to meet the delay constraints of real-time traffic in all network scenarios. Hence resource reservation and non-FIFO scheduling is required to adequately support real-time traffic.

VC Requirements

- Hybrid intra-subnet approaches should be supported. If a subnet supports a small number of multicast groups, each with only one or two local senders there is no significant difference between the VC requirements of the VC Mesh and MCS approaches. However the VC Mesh has better VC requirements than the MCS approach when a subnet supports many multicast groups, each with only one or two local senders. If the multicast group contains more than two multicast senders, the MCS approach should be employed.

- In terms of VC requirements, the Hop-by-Hop approach is better than all other inter-subnet delivery approaches, requiring at most one VC per link.
9. Conclusions

It also has the benefit that it is insensitive to the multicast group characteristics. Furthermore the VCs can be used for both unicast and multicast traffic.

- In environments where there are significantly more multicast senders than multicast groups, the Boundary approach has better VC requirements than the Cut-Through approach. However, when multicast groups only have one or two senders the Cut-Through approach VC requirements are lower, particularly at the edge of the network, since VCs do not need to be terminated at subnet routers.

- When multicast groups contain one or two senders, the Source Forwarding Tree approach requires less VCs per link than the Shared Forwarding Tree approach. However when multicast groups contain three or more senders the Shared Forwarding Tree approach has lower VC requirements.

Dynamic Multicast Group Support

- On average we found there is no significant difference between the MCS and VC Mesh approach either in terms of addition delay or resource requirements. This is contrary to commonly held hypotheses that the VC Mesh approach has significantly longer signalling network delays than the MCS approach. We found that the MCS approach only out-performs the VC Mesh approach when hundreds of senders or receivers are added per second, or the signalling network is highly utilised for some other reason.

- The Cut-Through approach can take significantly longer to add a new sender compared to the Hop-by-Hop approach, particularly when employed in conjunction with a Source Forwarding Tree. This is the case even when the multicast group contains only ten receivers. When multicast groups contain more receivers this difference is even more significant. The only circumstances we found where this did not hold is if senders forward large packets (larger than 10 kbytes). However, even in this situation, although the delay performance of the Cut-Through approach is superior, its signalling resource requirements are far greater, particularly when new receivers are being added. Hence we do not recommend the use of the Cut-Through approach.

- If new senders or receivers join subnets that already contain other multicast group members, the Boundary approach provides the best addition delay
9. Conclusions

performance. However the delay difference between the Boundary and Hop-by-Hop approaches is negligible, and the signalling resource requirements are identical. When adding new hosts to subnets that do not have any other multicast group members, the Hop-by-Hop approach has significantly better performance than the Boundary approach. Hence, the Hop-by-Hop approach is the preferred inter-subnet approach for supporting dynamic multicast groups. The Boundary approach should only be employed if the majority of new hosts join subnets that already have other multicast group members.

- The Shared Forwarding Tree approach provides significant benefits compared to the Source Forwarding Tree approach, when employed in conjunction with the Cut-Through or Boundary approaches. In the case of adding senders, Shared Forwarding Tree approaches provide significantly lower addition delays and signalling resource requirements. When adding receivers, the delay difference is often insignificant. However, there is an order of magnitude difference in the signalling resource requirements of the two forwarding tree approaches. Hence the Shared Forwarding Tree approach should be employed when supporting dynamic multicast groups. Note, when employed in conjunction with the Hop-by-Hop approach there is no significant difference between Source and Shared Forwarding Trees.

9.2.3 Overall

Overall, our unicast delivery analysis has shown that in most network environments the Cut-Through approach produces significantly lower delays than the Hop-by-Hop approach even when the Cut-Through VC must be created. Furthermore, the Buffered approach should be used to create Cut-Through VCs. The only circumstances where the Cut-Through approach does not provide significantly lower delays are: if the Cut-Through VC does not exist and the signalling network is highly utilised; or small UDP messages are being transmitted. If the signalling network is highly utilised, we recommend that the Hop-by-Hop approach be employed rather than attempting to create the VC. From a practical perspective an operator could monitor the signalling load on the end-end path, and hence know when to resort to the Hop-by-Hop approach, via a combination of timers and tracking the expected and current distributions of VC creation times.

Our unicast delivery analysis has shown that although the Hop-by-Hop approach has lower signalling and VC requirements than the Cut-Through approach, the signalling and VC requirements of the Cut-Through approach can be accom-
modated by current ATM switching equipment, even for WWW traffic, currently, the most prevalent IP application. This analysis also indicated that the performance of the Cut-Through approach is highly dependent on the VC holding time, and gives an indication of holding time values based on the relative costs of maintaining a VC and creating a new VC. Given the delay benefits of the Cut-Through approach it is the recommended approach for unicast delivery.

In terms of delay performance only, we also found that the Cut-Through, Source Forwarding Tree approach is the optimal multicast delivery system. However, in the majority of network scenarios there is no significant performance difference between the Boundary VC Mesh, and the Cut-Through approaches, or between Source and Shared Forwarding Trees.

The analysis also highlighted the importance of engineering the MCS and RP link and processor capacity to accommodate the expected traffic volume. If these are correctly engineered the difference between Source and Shared Forwarding Trees, and VC Mesh and MCS intra-subnet approaches becomes less significant. Our multicast delay analysis also indicated that none of the multicast delivery approaches were able to meet the delay constraints of real-time traffic in all network scenarios. Hence resource reservation and non-FIFO scheduling is required to adequately support traffic with strict QoS requirements.

In terms of VC requirements our studies have shown that the Hop-by-Hop approach is far superior to all other inter-subnet multicast delivery approaches requiring at most one VC per link. However the benefits of the Hop-by-Hop approach will decrease when providing QoS guarantees since this will mean multiple VCs will need to be created per link, at least one for each traffic class required.

The analysis also highlighted the sensitivity of multicast delivery approach VC requirements to the multicast group characteristics. When the network supports a small number of multicast groups with a small number of senders there is no significant difference between the Cut-Through and Boundary approaches, or the MCS and VC Mesh approaches. However when the network supports a large number of multicast groups each with only one or two senders, the Cut-Through and VC Mesh approaches have lower VC requirements than the Boundary and MCS approaches respectively. In contrast, if the network contains multicast groups with more than three senders, the Boundary and MCS approaches have lower VC requirements than the Cut-Through and VC Mesh approaches. A similar trade-off also occurs between Source and Shared Forwarding Trees. The analysis also highlighted the value of multiplexing traffic at RPs or local MCSs in terms of reducing VC requirements on receivers. If either of these approaches are employed, receivers only need to terminate one VC per multicast group they join. In contrast
other approaches require them to terminate up to one VC per sender. Hence there is a place for hybrid delivery schemes, where the choice of scheme depends on the characteristics of the multicast group.

This thesis also compared multicast delivery approaches when supporting dynamic multicast groups. The analysis clearly showed that regardless of whether adding new senders or receivers, the Cut-Through, Single Subnet, Source Forwarding Tree approach should not be employed for dynamic multicast groups. This is because of the large addition delays, the sensitivity of this approach to addition rate, and the heavy demands it places on the signalling network.

If new senders and receivers are added to subnets that already contain multicast group members, the Boundary approach provides the best dynamic multicast group support. However, the Hop-by-Hop approach provides only slightly higher delays than the Boundary approach. Moreover, when adding new hosts to subnets that do not contain multicast group members, the Hop-by-Hop provides significantly lower addition delays and lower demands on the signalling network. Hence, the Hop-by-Hop approach is the best inter-subnet delivery approach when supporting dynamic multicast groups.

This thesis has shown there is no significant difference between the dynamic multicast group support of the MCS and VC Mesh intra-subnet approaches, however there can be significant differences between Source and Shared Forwarding Tree performance, when employed with the Boundary or Cut-Through approaches. Shared Forwarding Trees provide significantly lower addition delays particularly when adding senders. Moreover, the signalling resource requirements are significantly lower when the Shared Forwarding Tree approach is employed, especially when adding receivers. However, when employed with the Hop-by-Hop approach, there is no significant difference between the two forwarding tree approaches.

To summarise this thesis has emphasised there is a trade-off between multicast delivery approaches depending on whether delay performance or resource requirements are of greatest concern. The analysis has also emphasised the importance of adequately dimensioning the network, particularly if the MCS intra-subnet approach or Shared Forwarding Tree approach is employed.

Given the trade-off between delay performance and resource requirements, we see an important role for hybrid delivery systems. This also falls in line with the multi-services platform principles of both Internet and ATM technologies. Internet applications and users of these applications have a wide range of services requirements, and may be willing to pay different amounts for these services. This thesis has clearly shown that the Hop-by-Hop approach is far superior to all other delivery approaches in terms of its resource requirements. Therefore, the Hop-
by-Hop approach should be employed as the basic service for carrying IP traffic in wide area ATM networks. However, this thesis has also clearly shown that the Cut-Through and Boundary approaches can provide significant delay savings compared to the Hop-by-Hop approach. In most network scenarios, the Boundary approach has delay characteristics resembling the Cut-Through approach, yet resource requirements that are significantly lower. Hence, we see an important role for these approaches, especially the Boundary approach, when providing premium grade services.

Moreover our analysis has highlighted that although the Cut-Through approach, and to a lesser extent, the Boundary approach have significantly higher resource requirements than the Hop-by-Hop approach, a lot of the time they are still within the capabilities of existing ATM equipment. This is particularly true if Shared Forwarding Trees are employed rather than Source Forwarding Trees. Hence the Cut-Through and Boundary approaches can be employed within operators networks. This brings into question the need for new lightweight signalling protocols promoted by many Label Switching techniques. They may have had a place when the signalling capabilities of ATM switches were limited, but now switches can support several hundred VC setups/s traditional signalling techniques may be sufficient. This is an area that requires further research.

Recommendations to operators when selecting an IP over ATM delivery mechanism are detailed below.

### 9.3 Recommendations

- The Hop-by-Hop approach should be employed as the basic delivery service for both unicast and inter-subnet multicast delivery.

- Given the mixed services environment of the Internet, we recommend that premium grade services be based on the Boundary delivery approach. This is because it provides significant delay savings compared to the Hop-by-Hop approach, in most environments it requires significantly lower resources than the Cut-Through approach, and lastly it does not depend on subnets at the edge of the core network being ATM based.

- When the Boundary or Cut-Through approaches are employed for unicast delivery, the Buffered approach should be used to create direct VCs.

- The selection of an optimal VC holding time is critical when employing the Boundary or Cut-Through approaches for unicast delivery. If the cost of
9. Conclusions

maintaining and creating a VC are equal, a holding time of five minutes is recommended.

- A hybrid multicast intra-subnet delivery approach is recommended. The VC Mesh approach should be employed for each group that has one or two local senders. The MCS approach should be employed for any active multicast groups that have three or more local senders.

- If the MCS intra-subnet approach is employed, it is crucial that the link bit rates and the processor capacity of the MCS are engineered to handle the expected traffic volume. If this is not the case MCS approach performance deteriorates substantially.

- If the Shared Forwarding Tree approach is employed, the RP should be placed as close as possible to the wire-line centre of the network. Furthermore, the RP processor capacity and the bit rate on the links connecting the RP to the rest of the network must be engineered to support the traffic volume.

- The multicast delivery analysis showed that the real time requirements of traffic flows can not be met by best-effort VCs even if the Cut-Through approach is employed. Hence resource reservation and non-FIFO scheduling is required to adequately support traffic with strict QoS requirements.

9.4 Future Work

This thesis has addressed the lack of performance analysis of mechanisms for carrying Internet traffic over ATM WANs. However there are still areas that require further analysis. Three of these are described below.

- Throughout this thesis we have assumed direct VCs are created via traditional ATM signalling protocols. In the literature a multitude of Label Switching approaches have been proposed that differ markedly in how they create VCs. Our findings have shown that the VC and signalling network requirements of the Boundary and Cut-Through approaches can be met by current ATM switching equipment in most network environments. However, there is a need to analyse these alternative Label Switching approaches in more detail, both in terms of their support for unicast traffic, and their support for dynamic multicast groups.
This thesis has considered delay performance and resource requirements of alternative delivery approaches for unicast and multicast Internet traffic over ATM networks in a zero loss network. This will have little impact on the relative performance difference of approaches when UDP is employed. However, when TCP is employed, the results presented in this thesis show the minimum difference performance operators can expect between approaches. In practice, the importance of selecting an approach with lower end-end delays will be even greater than indicated by the analysis in this thesis. Therefore areas for further study include the performance of approaches in the presence of loss, and a comparison of their buffer requirements.

This thesis has assumed that all traffic is carried in a best effort fashion, on UBR ATM VCs. UBR is the most commonly employed ATM service class for carrying Internet traffic. However, alternative service classes such as ABR have also been defined for carrying Internet traffic. The performance of delivery approaches with ABR style VCs is an area that requires further study. Furthermore, this thesis has shown that it is not possible to meet delay and jitter guarantees by carrying all traffic in a best effort fashion. Therefore, the performance of delivery approaches, when the network provides QoS guarantees to some application traffic also requires further research.

This thesis has considered the extra resource requirements of the Cut-Through approach when creating VCs. However the analysis has not considered operating scenarios where there are network failures (e.g. an ATM link fails). In general, approaches that employ multiple VCs on a link will require more resources to recover than an approach like the Hop-by-Hop approach which employs one VC per link. However further analysis is required to compare and determine the significance of differences in the processing resource requirements of approaches in the case of failure recovery.
Bibliography


BIBLIOGRAPHY


BIBLIOGRAPHY


Y Rekhter and J Halpern. NHRP for Destinations off the NBMA Subnetwork. IETF Internet Draft (Work in Progress), June 1998.


T Smith and G Armitage. IP Broadcast over ATM Networks. IETF Standards Track RFC 2226, October 1997.


[TA96b] R Talpade and M H Ammar. VC Consumption in the MARS Architecture for IP Multicast over ATM. In Workshop on Integration of IP+ATM Networks, November 1996.


BIBLIOGRAPHY


