

# **QoS Multicast for DiffServ on MPLS and IP Platforms**

Abdullah Ahmed Al Wehaibi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements for the  
Degree of Doctor of Philosophy at  
Concordia University  
Montreal, Quebec, Canada

November 2003

© Abdullah Al Wehaibi, 2003



National Library  
of Canada

Bibliothèque nationale  
du Canada

Acquisitions and  
Bibliographic Services

Acquisitons et  
services bibliographiques

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 0-612-90372-9*  
*Our file* *Notre référence*  
*ISBN: 0-612-90372-9*

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this dissertation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de ce manuscrit.

While these forms may be included in the document page count, their removal does not represent any loss of content from the dissertation.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

**Canada**

**CONCORDIA UNIVERSITY  
SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: **Abdullah Al Wehaibi**

Entitled: **QoS Multicast for DiffServ on MPLS and IP Platforms**

and submitted in partial fulfillment of the requirements for the degree of

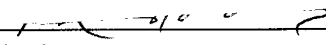
**DOCTOR OF PHILOSOPHY (Electrical & Computer Science)**


complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

\_\_\_\_\_ Chair  
Dr. B. Desai 

\_\_\_\_\_ External Examiner  
~~Dr. H. Mouftah~~ 

\_\_\_\_\_ External to Program  
Dr. M. Kadoch 

\_\_\_\_\_ Examiner  
Dr. S. Tahar 

\_\_\_\_\_ Examiner  
Dr. A. Sebak 

\_\_\_\_\_ Examiner  
Dr. A. Agarwal 

\_\_\_\_\_ Thesis Supervisor  
Dr. A.K. El-Hakeem 

Approved by \_\_\_\_\_  
Dr. M.K. Mehmet Ali, Graduate Program Director

**NOV 19 2003** 2003

\_\_\_\_\_  
Dr. N. Esmail, Dean  
Faculty of Engineering & Computer Science

# **ABSTRACT**

## **QoS Multicast for DiffServ on MPLS and IP Platforms**

**Abdullah Al Wehaibi, Ph.D.**  
**Concordia University, 2003**

Multicasting has become increasingly important with the emergence of Internet-based applications such Internet protocol (IP) telephony, audio/video conferencing, distributed databases and software upgrading. IP Multicasting is an efficient way to distribute information from a single source to multiple destinations at different locations. One of the challenges the Internet is facing today is to keep the packet forwarding performance up with the skyrocketing demand for bandwidth. On the other hand, the MultiProtocol Label Switching (MPLS), which is an Internet Engineering Task Force (IETF) framework, combines the flexibility of layer 3 routing and layer 2 switching, which enhances network performance in terms of scalability, computational complexity, latency and control message overhead. Besides, MPLS offers a vehicle for enhanced network services such as Quality of Services (QoS)/ Class of Service (CoS), Traffic Engineering and Virtual Private Networks (VPNs).

In this thesis, we present a new Fair Share Policy (FSP), which is a traffic policing mechanism that utilizes Differentiated Services (DiffServ) to solve the problems of QoS and congestion control. We compare the QoS performance of IP and MPLS multicasting, given their particular constraints. In order to achieve the required QoS, different techniques of reliable multicasting are adapted, such as Forward Error Correction (FEC),

Automatic Repeat Request (ARQ) or Hybrid FEC/ARQ with multicast or unicast repairs mechanisms so as to mitigate the effect of errors as well as packet loss. This reliable multicast is for both IP and MPLS platforms with Diffserv. Analytical and simulation models are suggested and employed.

The results provide insights into the comparisons between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ is adopted and when reliable multicast is considered. This comparison will be based on the following QoS measures: total packet delay, delay jitter and residual packet loss probability. Analysis and simulation tools are used to evaluate our fair share policy (FSP) for different homogeneous (when all routers are identical in their capabilities) and heterogeneous (when routers have different capabilities) network scenarios.

# ACKNOWLEDGEMENTS

I would like to take this opportunity to express my deep thanks and respect to my thesis supervisor, Dr. Ahmed ElHakeem. Dr. ElHakeem gives me his attention, advice, guidance and encouragement whenever I needed them. I can talk to him whenever I need, day or night remotely or face to face, with or without appointments. With his support and technical assistance, I was able to publish many conference and journal papers. With his help and kind revisions, I was able to make the first step to be a reviewer of famous journals such as Kluwer Telecommunications System and IEEE Communications Letters.

Also, I would like to express my great thanks to the examining committee for their constructive comments.

I will never forget these people, Dr. Mansour Al-Suliman, my supervisor during the B.Sc. degree who always encouraged me to continue my Ph.D.; Mr. Mohammed Al-Sohaili, who helped to get my scholarship; and all my friends who supported me a lot and wished for me a great success.

Finally, special thanks to my parents for their love, continuous support and prayer. I am in a real debit to my family members: my wife, and my lovely children: Ahmed, Fajer, Wahaj and Abdulaziz. My wife contributes significantly to the completion of this thesis. She is emotionally supportive while encouraging me to get things done. She did a lot of work so I can concentrate on my thesis. Thanks for love, support, patience and prayer.

**To My Beloved Parents  
To My Other Half- My Wife  
To My Lovely Kids: Ahmed, Fajer,  
Wahaj and Abdulaziz**

# TABLE OF CONTENTS (ToC)

<b>LIST OF FIGURES (LoF)</b> .....	<b>xii</b>
<b>LIST OF TABLES (LoT)</b> .....	<b>xix</b>
<b>LIST OF SYMBOLS (LoS)</b> .....	<b>xx</b>
<b>LIST OF ABBREVIATIONS (LoA)</b> .....	<b>xxiii</b>
<b>CHAPTER 1 INTRODUCTION</b> .....	<b>1</b>
1.1 Introduction to Thesis Problem .....	1
1.2 Thesis Objectives .....	4
1.3 Thesis Contribution .....	4
1.4 Thesis Approach (Methodology) .....	6
1.5 Thesis Organization .....	6
<b>CHAPTER 2 MULTICAST PRINCIPLES AND QOS</b> .....	<b>8</b>
2.1 Introduction .....	8
2.2 Key Concepts of MPLS .....	8
2.2.1 What is MPLS? .....	8
2.2.2 Label Assignment Rules .....	10
2.2.3 Label Swapping .....	11
2.3 MPLS multicasting .....	13
2.3.1 Aggregation and Granularity .....	14
2.3.2 Flooding and Pruning .....	14
2.3.3 Source and Shared Trees .....	15
2.3.4 Label Switched Patch (LSP) Triggers .....	16



2.3.5 Label Advertisement .....	17
2.3.6 Types of MPLS routing .....	17
2.4 MPLS Multicast Considerations .....	18
2.4.1 Label Switched Path (LSP) Establishment Latency .....	18
2.4.2 Limited Label Space .....	18
2.4.3 Security .....	19
2.5 Quality of Service (QoS) in the Internet .....	19
2.5.1 Integrated services and Resource ReSerVation Protocol (RSVP) Model .....	20
2.5.2 Differentiated Services (DiffServ) Model .....	20
2.5.2.1 DiffServ Architecture .....	21
2.6 Reliable Multicast .....	24
2.6.1 Introduction to Error Control .....	24
2.6.2 Error Control Strategies .....	24
2.6.2.1 Automatic Repeat Request (ARQ) .....	24
2.6.2.2 Forward Error Correction (FEC) .....	26
2.6.2.3 Hybrid FEC/ARQ .....	28
2.6.3 Reed-Solomon Codes .....	28
2.6.3.1 Properties of Reed-Solomon Codes .....	28
2.6.3.3 Reed-Solomon Decoding .....	29
2.7 Conclusions .....	30
<b>CHAPTER 3 RELATED WORK OF MULTICASTING TECHNOLOGIES</b> .....	<b>31</b>
3.1 Introduction .....	31
3.2 Congestion Control and Traffic Policing .....	31

3.3 Reliable IP Multicast .....	34
3.4 DiffServ Multicasting .....	41
3.5 MPLS Multicasting .....	44
3.6 DiffServ/MPLS Multicasting .....	45
3.7 QoS in Heterogeneous Networks .....	49
3.8 Conclusions .....	49
<b>CHAPTER 4 QOS MULTICAST FOR DIFFSERV OVER MPLS AND IP HOMOGENEOUS NETWORKS .....</b>	<b>51</b>
4.1 Introduction .....	51
4.2 The Analytical Model Underlying the Fair Share Policy (FSP) .....	53
4.2.1 Analysis Results .....	60
4.3 Homogeneous Reliable Multicast Tree .....	68
4.3.1 The Analytical Model Underlying the Fair Share Policy (FSP) .....	68
4.3.2 Reliable Multicast Cases Under Study .....	69
4.3.3 Analysis Results .....	82
4.4 Conclusions .....	92
<b>CHAPTER 5 RESIDUAL PACKET LOSS PROBABILITY FOR DIFFSERV OVER IP AND MPLS MULTICAST TREES .....</b>	<b>94</b>
5.1 Introduction .....	94
5.2 Residual Packet Loss Probability Calculations Using Worst Case and Approximate Methods .....	96

5.3 Residual Packet Loss Probability Calculation Using an Exact Method .....	99
5.4 Analysis Results .....	102
5.5 Conclusions .....	107
<b>CHAPTER 6 RELIABLE QOS MULTICAST FOR DIFFSERV OVER</b>	
<b>HETEROGENEOUS NETWORKS</b> .....	<b>109</b>
6.1 Introduction .....	109
6.2 Reliable QoS Multicast for DiffServ Over Heterogeneous MPLS Networks ....	109
6.2.1 Heterogeneous MPLS networks .....	110
6.2.2 The Analytical Model Underlying the Fair Share Policy (FSP) .....	112
6.2.3 Reliable Multicast Cases Under Study .....	114
6.2.4 Analysis Results .....	122
6.3 Reliable QoS Multicast for DiffServ Over Hybrid FEC/ARQ IP and MPLS	
Heterogeneous Networks .....	130
6.3.1 Heterogeneous MPLS and Heterogeneous IP based networks .....	131
6.3.2 The Analytical Model Underlying the Fair Share Policy (FSP) .....	133
6.3.3. Reliable FEC/ARQ Multicast Case using Unicast Repair Packets ....	134
6.3.4 Analysis Results .....	139
6.4 Conclusions .....	144
<b>CHAPTER 7 SIMULATION OF FEC/AQR MULTICAST FOR DIFFSERV</b>	
<b>OVER MPLS AND IP PLATFORMS</b> .....	<b>147</b>
7.1 Introduction .....	147
7.2 The Simulation Model Underlying the Fair Share Policy (FSP) .....	148
7.2.1 Simulation Model Assumptions .....	149

7.2.2 Input Buffer Characteristics .....	150
7.2.3 Server (Sender) Characteristics .....	151
7.2.4 IP and MPLS Source Arrivals .....	151
7.2.5 Interleaving .....	152
7.3 Performance Measures .....	152
7.4 Homogeneous IP/MPLS Multicast Networks .....	154
7.4.1 Simulation Programs .....	154
7.4.2 Description of the Simulation Programs .....	154
7.4.3 The Simulation Modules .....	154
7.4.4 FEC/ARQ Operation .....	155
7.4.5 Simulation Results .....	159
7.5 Heterogeneous MPLS Multicast Networks .....	163
7.5.1 Different Source Arrivals .....	164
7.5.2 Simulation Programs .....	165
7.5.3 Simulation Results .....	168
7.6 Comparison of Analysis and Simulation Results .....	170
7.7 Validity of the Simulation Programs .....	173
7.8 Conclusions .....	173
<b>CHAPTER 8 THESIS CONCLUSIONS AND FUTURE WORK .....</b>	<b>175</b>
8.1 Thesis Conclusions .....	175
8.2 Suggested Future Work .....	179
<b>REFERENCES .....</b>	<b>181</b>

## LIST OF FIGURES (LoF)

Fig. 2-1 MPLS header format .....	10
Fig. 2-2 Label swapping and forwarding .....	13
Fig. 2-3 Flooding and pruning .....	15
Fig. 2-4 Source and shared trees .....	16
Fig. 2-5 Differentiated Services (DS) field structure .....	22
Fig. 2-6 Packet classifier and scheduler .....	22
Fig. 2-7 FEC operation .....	27
Fig. 2-8 Reed-Solomon Code word .....	29
Fig. 3-1 Token Bucket Scheme .....	33
Fig. 3-2 Exponentially Weighted Moving Average Scheme .....	33
Fig. 3-3 NRS Problem in DiffServ multicast .....	42
Fig. 4-1 The analytical model .....	55
Fig. 4-2 The coupled discrete Markovian state diagrams .....	56
Fig. 4-3 Expected number of packets in the buffer for all sources for both IP and MPLS (small $\tau$ ) .....	63
Fig. 4-4 Packet loss probability for all sources for both IP and MPLS (small $\tau$ ) .....	63
Fig. 4-5 Expected number of packets in the buffer for all sources for both IP and MPLS (large $\tau$ ) .....	64
Fig. 4-6 Packet loss probability for all sources for both IP and MPLS (large $\tau$ ) .....	64
Fig. 4-7 Expected number of packets in the buffer for all sources for both IP and MPLS (effect of MPLS factor) .....	64
Fig. 4-8 Packet loss probability for all sources for both IP and MPLS (effect of MPLS factor) .....	64

Fig. 4-9 IP Expected number of packets in the buffer for source 1 versus IP factor and processing factor (high arrival rates) .....	65
Fig. 4-10 IP Expected number of packets in the buffer for source 2 versus IP factor and processing factor (high arrival rates) .....	65
Fig. 4-11 MPLS Expected number of packets in the buffer for source 1 versus IP factor and MPLS factor (high arrival rates) .....	65
Fig. 4-12 MPLS Expected number of packets in the buffer for source 2 versus IP factor and MPLS factor (high arrival rates) .....	65
Fig. 4-13 IP Packet loss probability for source 1 versus IP factor and processing factor (high arrival rates) .....	66
Fig. 4-14 IP Packet loss probability for source 3 versus IP factor and processing factor (high arrival rates) .....	66
Fig. 4-15 MPLS Packet loss probability for source 1 versus IP factor and MPLS factor (high arrival rates) .....	66
Fig. 4-16 MPLS Packet loss probability for source 3 versus IP factor and MPLS factor (high arrival rates) .....	66
Fig. 4-17 IP Expected number of packets in the buffer for source 1 versus IP factor and processing factor (low arrival rates) .....	67
Fig. 4-18 MPLS Expected number of packets in the buffer for source 1 versus IP factor and MPLS factor (low arrival rates) .....	67
Fig. 4-19 IP Packet loss probability for source 3 versus IP factor and processing factor (low arrival rates) .....	67
Fig. 4-20 MPLS Packet loss probability for source 3 versus IP factor and MPLS factor (low arrival rates) .....	67
Fig. 4-21 A complete homogeneous binary multicast tree .....	68
Fig. 4-22 Conditional probabilities tree for unicast repair .....	76
Fig. 4-23 Flowchart of program which calculate performance measures of Hybrid FEC/ARQ unicast repairs .....	80
Fig. 4-24 Total packet delay versus IP factor (No FEC or ARQ and small $\tau$ ) .....	85
Fig. 4-25 Total delay jitter versus IP factor (No FEC or ARQ and small $\tau$ ) .....	85

Fig. 4-26 Residual loss probability versus IP factor (No FEC or ARQ and small $\tau$ )	85
Fig. 4-27 Total packet delay versus IP factor (No FEC or ARQ and large $\tau$ )	85
Fig. 4-28 Total delay jitter versus IP factor (No FEC or ARQ and large $\tau$ )	86
Fig. 4-29 Residual loss probability versus IP factor (No FEC or ARQ and large $\tau$ )	86
Fig. 4-30 Total packet delay versus MPLS factor (No FEC or ARQ)	86
Fig. 4-31 Total delay jitter versus MPLS factor (No FEC or ARQ)	86
Fig. 4-32 Residual loss probability versus MPLS factor (No FEC or ARQ)	87
Fig. 4-33 Total packet delay versus IP factor (FEC only and large $\tau$ )	87
Fig. 4-34 Total delay jitter versus IP factor (FEC only and large $\tau$ )	87
Fig. 4-35 Residual loss probability versus IP factor (FEC only and large $\tau$ )	87
Fig. 4-36 Total packet delay versus MPLS factor (FEC only)	88
Fig. 4-37 Total packet delay versus IP factor (ARQ multicast and large $\tau$ )	88
Fig. 4-38 Total delay jitter versus IP factor (ARQ multicast and large $\tau$ )	88
Fig. 4-39 Residual loss probability versus IP factor (ARQ multicast and large $\tau$ )	88
Fig. 4-40 Total delay jitter versus MPLS factor (ARQ multicast)	89
Fig. 4-41 Total packet delay versus IP factor (ARQ unicast and large $\tau$ )	89
Fig. 4-42 Total delay jitter versus IP factor (ARQ unicast and large $\tau$ )	89
Fig. 4-43 Residual loss probability versus IP factor (ARQ unicast and large $\tau$ )	89
Fig. 4-44 Residual loss probability versus MPLS factor (ARQ unicast)	90
Fig. 4-45 Total packet delay versus IP factor (Hybrid multicast and large $\tau$ )	90
Fig. 4-46 Total delay jitter versus IP factor (Hybrid multicast and large $\tau$ )	90
Fig. 4-47 Residual loss probability versus IP factor (Hybrid multicast and large $\tau$ )	90
Fig. 4-48 Total packet delay versus MPLS factor (Hybrid multicast)	91
Fig. 4-49 Total packet delay versus IP factor (Hybrid unicast and large $\tau$ )	91

Fig. 4-50 Total delay jitter versus IP factor (Hybrid unicast and large $\tau$ )	91
Fig. 4-51 Residual loss probability versus IP factor (Hybrid unicast and large $\tau$ )	91
Fig. 4-52 Residual loss probability versus MPLS factor (Hybrid unicast)	92
Fig. 5-1 A complete homogeneous binary multicast tree	96
Fig. 5-2 The 4 trials tree of success to repair 1 error for certain priority traffic $p$	99
Fig. 5-3 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=31)	104
Fig. 5-4 Residual Loss Probability for IP and MPLS source 2 versus IP factor (N=31)	104
Fig. 5-5 Residual Loss Probability for IP and MPLS source 3 versus IP factor (N=31)	105
Fig. 5-6 Residual Loss Probability for IP and MPLS source 2 versus MPLS factor (N=15)	105
Fig. 5-7 Residual Loss Probability for IP and MPLS source 1 versus MPLS factor (N=15)	105
Fig. 5-8 Residual Loss Probability for IP and MPLS source 2 versus MPLS factor (N=15)	105
Fig. 5-9 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=7)	106
Fig. 5-10 Residual Loss Probability for IP and MPLS source 3 versus IP factor (N=7)	106
Fig. 5-11 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=63)	106
Fig. 5-12 Residual Loss Probability for IP and MPLS source 2 versus IP factor (N=63)	106
Fig. 5-13 Residual Loss Probability for IP and MPLS source 2 versus IP factor (N=63)	107
Fig. 6-1 1 IP router at level 5 in a heterogeneous MPLS network	112
Fig. 6-2 1 IP router at level 4 in a heterogeneous MPLS network	112



Fig. 6-3 Flowchart of program which calculate performance measures of Hybrid FEC/ARQ unicast repairs .....	121
Fig. 6-4 Total packet delay versus IP factor (No FEC or ARQ and small $\tau$ ) .....	124
Fig. 6-5 Total delay jitter versus IP factor (No FEC or ARQ and small $\tau$ ) .....	124
Fig. 6-6 Residual loss probability versus IP factor (No FEC or ARQ and small $\tau$ ) .....	125
Fig. 6-7 Total packet delay versus IP factor (FEC only and small $\tau$ ) .....	125
Fig. 6-8 Total delay jitter versus IP factor (FEC only and small $\tau$ ) .....	125
Fig. 6-9 Residual loss probability versus IP factor (FEC only and small $\tau$ ) .....	125
Fig. 6-10 Total packet delay versus IP factor (ARQ multicast and small $\tau$ ) .....	126
Fig. 6-11 Total delay jitter versus IP factor (ARQ multicast and small $\tau$ ) .....	126
Fig. 6-12 Residual loss probability versus IP factor (ARQ multicast and small $\tau$ ) .....	126
Fig. 6-13 Total packet delay versus IP factor (ARQ unicast and small $\tau$ ) .....	126
Fig. 6-14 Total delay jitter versus IP factor (ARQ unicast and small $\tau$ ) .....	127
Fig. 6-15 Residual loss probability versus IP factor (ARQ unicast and small $\tau$ ) .....	127
Fig. 6-16 Total packet delay versus IP factor (hybrid multicast and small $\tau$ ) .....	127
Fig. 6-17 Total delay jitter versus IP factor (hybrid multicast and small $\tau$ ) .....	127
Fig. 6-18 Residual loss probability versus IP factor (hybrid multicast and small $\tau$ ) .....	128
Fig. 6-19 Total packet delay versus IP factor (hybrid unicast and small $\tau$ ) .....	128
Fig. 6-20 Total delay jitter versus IP factor (hybrid unicast and small $\tau$ ) .....	128
Fig. 6-21 Residual loss probability versus IP factor (hybrid unicast and small $\tau$ ) .....	128
Fig. 6-22 Heterogeneous binary multicast tree with 2 DRs and 2 Subnets .....	133
Fig. 6-23 Heterogeneous binary multicast tree with 3 DRs and 3 Subnets .....	133
Fig. 6-24 Total packet delay for all IP sources versus IP factor (homogeneous networks) .....	141
Fig. 6-25 Total packet delay for MPLS sources versus IP factor (homogeneous networks) .....	141

Fig. 6-26 Residual loss probability for all IP sources versus IP factor (homogeneous networks) .....	141
Fig. 6-27 Residual loss probability for MPLS sources versus IP factor (homogeneous networks) .....	141
Fig. 6-28 Total packet delay for all IP sources versus IP factor (heterogeneous networks) .....	142
Fig. 6-29 Total packet delay for MPLS sources versus IP factor (heterogeneous networks) .....	142
Fig. 6-30 Residual loss probability for all IP sources versus IP factor (heterogeneous networks) .....	142
Fig. 6-31 Residual loss probability for MPLS sources versus IP factor (heterogeneous networks) .....	142
Fig. 6-32 Total packet delay for all IP sources versus MPLS factor (heterogeneous networks) .....	143
Fig. 6-33 Total packet delay for MPLS sources versus MPLS factor (heterogeneous networks) .....	143
Fig. 6-34 Residual loss probability for all IP sources versus MPLS factor (heterogeneous networks) .....	143
Fig. 6-35 Residual loss probability for MPLS sources versus MPLS factor (heterogeneous networks) .....	143
Fig. 7-1 The Router's Simulation Model .....	149
Fig. 7-2 (a) Flowchart of the Simulation Program Part A (Initialization and Main Module) .....	157
Fig. 7-2 (b) Flowchart of the Simulation Program Part B (Event Generator and FEC/ARQ Operation) .....	158
Fig. 7-2 (c) Flowchart of the Simulation Program Part C (Event Scheduler) .....	159
Fig. 7-3 Total packet delay versus IP factor (small $\tau$ ) .....	161
Fig. 7-4 Total delay jitter versus IP factor (small $\tau$ ) .....	161
Fig. 7-5 Residual loss probability versus IP factor (small $\tau$ ) .....	161

Fig. 7-6 Total packet delay versus IP factor (large $\tau$ ) .....	161
Fig. 7-7 Total delay jitter versus IP factor (large $\tau$ ) .....	162
Fig. 7-8 Residual loss probability versus IP factor (large $\tau$ ) .....	162
Fig. 7-9 Total packet delay versus MPLS factor .....	162
Fig. 7-10 Total delay jitter versus MPLS factor .....	162
Fig. 7-11 Residual loss probability versus MPLS factor .....	163
Fig. 7-12 Flowchart of heterogeneous MPLS network configuration module	166-167
Fig. 7-13 Total packet delay for all sources versus MPLS factor .....	169
Fig. 7-14 Delay jitter for all sources versus MPLS factor .....	169
Fig. 7-15 Residual loss probability for all sources versus MPLS factor .....	170
Fig. 7-16 Total packet delay for all IP sources (homogeneous) .....	171
Fig. 7-17 Residual loss probability for all MPLS sources (homogeneous) .....	172
Fig. 7-17 Residual loss probability for all MPLS sources (heterogeneous) .....	172

## LIST OF TABLES (LoT)

Table 2-1 MPLS features .....	11
Table 6-1 The distribution table of the existence possibilities of 1 IP, 2 IP or 3 IP routers in a heterogeneous MPLS network with 31 routers .....	129
Table 6-2 The distribution table of the existence possibilities of 1 DR, 2 DR in addition to DR at root in a heterogeneous MPLS or IP network with 31 nodes .....	134

## LIST OF SYMBOLS (LoS)

$\alpha_1, \alpha_2$ and $\alpha_3$	Arrival probabilities for sources 1, 2 and 3 respectively.
$\beta_1, \beta_2$ and $\beta_3$	Service probabilities for sources 1,2 and 3 respectively.
$p$	Traffic source priority ( $p=1,2$ or $3$ ).
$\lambda_p$	Arrival rate for priority traffic $p$ .
$\Delta T$	Service time
$\max_p$	Maximum buffer size for priority traffic $p$ .
$B$	Total system (router) buffer size
$Pc_p$	Probability of successful delivery to next router for priority traffic $p$
$\alpha_p^i$	The intrinsic arrival probability for priority traffic $p$ .
$\tau$	IP processing factor
$\xi_1$	IP control overhead factor (IP factor)
$\xi_2$	MPLS control overhead factor (MPLS factor)
$\overline{E_p(n)}$	Expected number of packets in the buffer for priority traffic $p$ .
$Po_p$	Byte overflow (loss) probability for priority traffic $p$ .
$Pe_p$	Byte error probability for priority traffic $p$ .
$Ps_p$	Probability of multicast success for priority traffic $p$ .
$Ploss_p$	Residual packet loss probability for priority traffic $p$ .
$D_{pTotal}$	Total delay for packet with priority $p$ .
$D$	Multicast tree depth.
$\overline{D_p}$	Average packet delay per router for priority $p$ traffic.
$\sigma_{pTotal}$	Total delay jitter for priority traffic $p$ .
$L$	Number of bytes per packet

$N$	Total number of routers in the multicast network
$r_p$	FEC coding rate for priority traffic $p$ .
$k$	Number of original data symbols
$n$	Total number of symbols after applying FEC encoding
$e$	Number of FEC erasures.
$F_p$	Total number of failures for priority traffic $p$ .
$T_p$	Total number of ARQ trials for priority traffic $p$
$z$	Number of multicast trials of a certain packet with priority $p$ .
$\Delta$	The extra arrival rate due to processing of unicast repairs.
$Ps_{pw}$	Worst case probability of success for priority traffic $p$ .
$Ploss_{pw}$	Worst case residual packet loss probability for priority traffic $p$ .
$Ps_{pA}$	Approximate (average) probability of success for priority traffic $p$ .
$Ploss_{pA}$	Approximate (average) residual packet loss probability for priority traffic $p$
$Ploss_{pE}$	Exact residual packet loss probability for priority traffic $p$
$\xi_3$	MPLS with extra processing factor (or ME factor)
$\xi_4$	MPLS egress or ingress router with extra processing factor (or EI factor)
$Ps_i$	Probability of a situation.
$O_i$	Number of occurrences of situation $i$ .
$T_i$	Probability of the number of IP routers in situation $i$ .
$k$	Router type ( $k=1,2,3$ , and $4$ for IP, ME, EI and M routers respectively)
$\overline{\theta}_k$	Average probability of the number of IP routers in the multicast network.
$\overline{Pc}_p$	Average probability of successful delivery to next router for priority traffic $p$ (heterogeneous networks)

$\overline{\Delta D}_{i,p}$	The extra delay the packet of priority p endures from the providing DR to the current router requesting the repair packet for sub-tree i.
$D_i$	The depth of sub-tree i
$F_{i,p}$	The average number of failures in sub-tree i for priority traffic p
$D_{i,p}$	The packet delay per router in sub-tree i for priority traffic p.
$\overline{D}_{i,p}$	The average packet delay of sub-tree i for priority traffic p.
$Ph_i$	The probability that the current DR router has the repair packet
$dis_i$	Distance between the current DR and the parent DR.
$Pov_i$	Current DR probability of buffer overflow.
$\theta_{i,j}$	Probability that a router being under DR router (i) in situation j.
$O(j)$	Number of occurrences of situation j.
$S$	Total number of situations
$\overline{D}_{p,j}$	Situation j average packet delay for priority traffic p.
$\overline{D}_{i,p,j}$	Average packet delay of sub-tree (i) and for priority traffic p in situation j.
$\overline{Ps}_{p,j}$	Situation j average probability of success for priority traffic p.
$\overline{Ps}_{i,p,j}$	Average probability of success in the multicast sub-tree I for priority traffic p in the situation number j.

## **LIST OF ABBREVIATIONS (LoA)**

ACK	Acknowledgement
ARQ	Automatic Repeat Request
ATM	Asynchronous Transfer Mode
BGP	Border Gateway Protocol
BSC	Binary Synchronous Communications
CBT	Core Based Tree
CoS	Class of Service
CPE	Customer Premises Equipment
DiffServ	Differentiated Services
DLCI	Data Link Connection Identifier
DSCP	Differentiated Services Code Points
DVMRP	Distance Vector Multicast Routing Protocol
EWMA	Exponentially Weighted Moving Average
FEC	Forward Error Correction
FIFO	First In First Out
FR	Frame Relay
FSP	Fair Share Policy
HDLC	High-Level Data Link Control
IBM	International Business Machines
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IGMP	Internet Group Management Protocol



IntServ	Integrated Services
IP	Internet Protocol
ISO	International Standards Organization
ISP	Internet Service Provider
L2	Layer 2
L3	Layer 3
LAN	Local Area Network
LDP	Label Distribution Protocol
LER	Label Edge Router
LSP	Label Switched Path
LSR	Label Switching Router
MPLS	Multiprotocol Label Switching
MRT	Multicast Routing Table
NAK	Negative Acknowledgement
NRS	Neglected Reservation Subtree
OSI	Open Systems InterConnection
OSPF	Open Shortest Path First
PDU	Packet Data Unit
PHB	Per Hop Behavior
PIM-DM	Protocol Independent Multicast- Dense Mode
PIM-SM	Protocol Independent Multicast- Sparse Mode
QoS	Quality of Service
RS	Reed Solomon

RSVP	Resource reServation Protocol
SLA	Service Level Agreement
TB	Token Bucket
TCP	Transmission Control Protocol
TE	Traffic Engineering
TTL	Time- To-Live
UDP	User Datagram Protocol
VCI	Virtual Circuit Identifier
VPI	Virtual Path Identifier
VPN	Virtual Private Network
WAN	Wide Area Network
PSC	PHB Scheduling Class
EXP	Experimental bits

# **CHAPTER 1 INTRODUCTION**

## **1.1 Introduction to Thesis Problem**

One of the Key requirements in deploying a pervasive and ubiquitous information superhighway is the development of a global integrated communication infrastructure capable of physically moving user's information among geographically spread areas. This global communication network has to cope with the tremendous amount of traffic generated by a huge number of anticipated users. In addition to that, it has to deal with a wide range of traffic characteristics. This is because the network will have to support, simultaneously; applications having a wide range of expectations and requirements.

Among the networks available today, those based on the concept of packet switching, which offer the highest potential degree of flexibility to meet the different application requirements, and therefore offer the best available technology on which the global communication infrastructure could rely. Because it already connects millions of users, the Internet is the uncontested prime candidate to constitute the core of global infrastructure.

Group communication or more specifically multicasting has been at the center of interest in the area of Internet activities and has already contributed to some major successes. Multicasting has become increasingly important with the emergence of Internet-based applications such Internet protocol (IP) telephony, audio/video conferencing, distributed databases and software upgrading. IP Multicast supports this type of transmission by enabling sources to send a single copy of a message to multiple recipients at different locations who explicitly want to receive the information. This is an

efficient way than requiring the source to send an individual copy of a message to each requester (referred to as point-to-point unicast), in which case the number of receivers is limited by the bandwidth available to the sender. It is also more efficient than broadcasting one copy of the message to all nodes (broadcast) on the network, since many nodes may not want the message, and because broadcasts are limited to a single subnet.

Multicast is a receiver-based concept: receivers join a particular multicast session group and traffic is delivered to all members of that group by the network infrastructure. The sender does not need to maintain a list of receivers. Only one copy of a multicast message will pass over any link in the network, and copies of the message will be made only where paths diverge at a router. Thus, IP Multicast yields many performance improvements and conserves more bandwidth than other means such as unicast or broadcast.

One of the challenges the Internet is facing today is to keep the packet forwarding performance up with the skyrocketing demand for bandwidth. Not only the number of attached hosts keeps growing exponentially, but also the increasing popularity of multimedia applications inflates the amount of traffic that every individual host is generating.

In practice, IP is considered a layer 3 (L3) protocol. Recent developments in Multiprotocol Label Switching (MPLS) open new possibilities to address some of the limitations of IP systems. MPLS is an Internet Engineering Task Force (IETF) standard. It replaces the IP forwarding by a simple label lookup mechanism. MPLS combines the flexibility of layer 3 routing and layer 2 (L2) switching, which enhances network

performance in terms of scalability, computational complexity, latency and control message overhead. Besides this, MPLS offers a vehicle for enhanced network services such as Quality of Services (QoS)/ Class of Service (CoS), Traffic Engineering and Virtual Private Networks (VPNs).

IP multicast in MPLS networks is still an open issue. On the other hand, the IETF DiffServ working group is looking at a more scalable model and more likely to be easier to implement than IntServ/RSVP model. In the DiffServ architecture, traffic that requires the same Per-Hop-Behavior (PHB) is aggregated into a single queue. Packets are classified into the corresponding queues using their DiffServ Code Points (DSCP). Packets use DSCP bits in order to receive a particular PHB, or forwarding treatment. Marking, classification, traffic conditioning or policing are done at network boundaries (first router for example) and packet treatment and handling is carried on each network node.

In this thesis, different analytical and simulation models are employed. Each model represents a typical IP or MPLS router where the traffic policing mechanism fair share policy (FSP) process different independent sources corresponding to different input traffic classes. The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). Each router may have different capabilities; for example, one router could have the ability to correct errors (FEC) and use ARQ, one may use only ARQ but cannot correct errors, a third one may not have MPLS capability.

In this thesis, we present a new fair share policy (FSP) that utilizes Differentiated Services to solve the problems of QoS and congestion control when reliable multicasting

is used. Analysis and simulation tools are used to evaluate our new fair share policy (FSP) for different network scenarios (homogeneous and heterogeneous cases). The results provide insights for the comparisons between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ and reliability are adopted. Through out this thesis, we will use the term MPLS multicast to denote IP multicast in MPLS networks.

## **1.2 Thesis Objectives**

To compare the QoS performance of IP and MPLS multicasting, given their particular constraints. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. In this thesis, we evaluate the QoS performance measures such as total packet delay, packet loss probability and delay jitter of Diffserv classes (traffics with different priority classes) for both MPLS and IP platforms when reliable multicasting is used. This comparison would be carried for different homogeneous (when all routers are identical in their capabilities) and heterogeneous (when routers have different capabilities) network scenarios.

## **1.3 Thesis Contribution**

1- We present a new fair share policy (FSP) that utilizes Differentiated Services traffic to solve the problems of QoS and congestion control when reliable multicasting is used. FSP is not a call admission rather it is a traffic policing

mechanism. In FSP, packets are discarded in case of congestion differently at each queue according to source priority and the maximum number in the queue; i.e. the source with higher priority say real time voice and video will experience less packet discarding than sources with lower priorities. Moreover, FSP guarantees fairness among flows having the same priority (i.e., required QoS) in buffer space allocated to lower priority traffic say email or web browsing is larger; thus leading to less packet discard.

2- In order to achieve the required QoS, different techniques of reliable multicast will be adapted, such as Forward Error Correction (FEC) or Automatic Repeat Request (ARQ) or Hybrid FEC/ARQ with multicast or unicast repair mechanisms so as to mitigate the effect of errors as well as packet loss. This reliable multicast is used for both IP and MPLS platforms with Diffserv.

3- Analytical and simulation models are suggested and employed. A model represents a typical IP or MPLS router and FSP traffic policing mechanism process different independent sources corresponding to different input traffic classes. The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network).

4- Fine-tuning of various parameters of the reliable multicast in the environment above also considers the homogeneous network deployment where all assumed QoS measures are adapted on all routers. In a different scenario, we investigate the heterogeneous case where the QoS measures exist only on some of the routers.

5- Analysis and simulation tools are used to evaluate fair share policy for different scenarios. The results provide insights for the comparisons between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy.

6- To derive various conclusions and suggestions regarding the performance comparisons between MPLS multicast and IP multicast when DiffServ is accommodated for both homogeneous and heterogeneous network models.

The thesis will not address routing, rerouting, tree establishment issues.

## **1.4 Thesis Approach (Methodology)**

In this thesis, we carry out a comprehensive study to investigate the QoS measures for DiffServ on MPLS and IP Platforms when reliable multicasting is used for different network scenarios. This study is accomplished through analytical and simulation tools. Using these tools, various conclusions and insights are derived. Analytical tools are accurate tools for evaluating our network models; however sometimes many assumptions are made which make our analytical models valid for specific cases. On the other hand, popularity of the simulation programs stems from the fact that simulation makes it possible to systematically study a network to a desired level of details, when exact analysis is not feasible. Through simulation, it is easy to analyse network performance for different network dimensions and configurations.

## **1.5 Thesis Organization**

Chapter 2 presents a literature survey. In this chapter, the basic concepts of MPLS, MPLS multicast considerations, QoS in the Internet and reliable multicast will be addressed.



Chapter 3 discusses the existing analysis or simulation works which are related to our thesis. In this chapter, we will address the following subjects: reliable IP multicast, MPLS multicast, DiffServ multicast, MPLS/DiffServ multicast and QoS in heterogeneous networks.

Chapter 4 presents a comprehensive analysis comparison between IP and MPLS homogeneous multicast networks when both DiffServ and reliability are adopted. This comparison will be based on the following QoS measures: total packet delay, delay jitter and residual packet loss probability.

In chapter 5, we derive and compare three mathematical expressions, which can be used to calculate the residual packet loss probability in binary multicast trees for both IP and MPLS networks.

Chapter 6 presents a comprehensive analysis comparison between homogeneous IP, homogeneous MPLS, heterogeneous IP and heterogeneous MPLS multicast networks when both DiffServ and reliability are adopted. This comparison will be based on the following QoS measures: total packet delay, delay jitter and residual packet loss probability.

Chapter 7 addresses computer simulations. This chapter describes the simulation models, source traffic characteristics, server characteristics, simulation programs and the assumptions assumed during the simulation.

Thesis conclusions and suggestions for further work are given in the last chapter i.e. chapter 8.

# **CHAPTER 2 MULTICAST PRINCIPLES AND QOS**

## **2.1 Introduction**

IP Multicast enables sources to send a single copy of a message to multiple recipients at different locations [1-9]. Recent developments in Multiprotocol label Switching (MPLS) open new possibilities to address some of the limitations of IP systems. MPLS is an Internet Engineering Task Force (IETF) standard [10]. It replaces the IP forwarding by a simple label lookup mechanism. MPLS combines the flexibility of layer 3 (L3) routing and layer 2 (L2) switching, which enhances network performance in terms of scalability, computational complexity and latency. MPLS multicast is still an open issue. On the other hand, the IETF DiffServ working group is looking at a more scalable model and more likely to be easier to implement than IntServ/RSVP model.

Reliable multicasting is used to provide QoS in group communications for real time multimedia applications such as voice/video streaming. Two main error control strategies are well known. These are the FEC (Forward Error Correction) strategy and the ARQ (Automatic Repeat Request) strategy.

In this chapter, a survey of recent literature will be presented. The basic concepts of MPLS, MPLS multicast considerations, QoS in the Internet (mainly DiffServ) and reliable multicast will be addressed in this chapter.

## **2.2 Key Concepts of MPLS**

### **2.2.1 What is MPLS?**

Multiprotocol Label Switching (MPLS) is an Internet Engineering Task Force (IETF) specified framework [10]. MPLS is currently under standardization by IETF. MPLS is an

extension to the existing IP architecture [11]. It has become a key player in the emerging multi-service Internet.

In traditional networks, routers make independent forwarding decisions for each packet traveling through the network. Each router independently chooses a next hop for the packet, based on its analysis of the packet's header and the results of running the routing algorithm. Choosing the next hop can therefore be thought of as the composition of two functions. The first function partitions the entire set of possible packets into a set of "*Forwarding Equivalence Classes (FECs)*". The second maps each FEC to a next hop. Insofar as the forwarding decision is concerned, different packets, which get mapped into the same FEC, are indistinguishable. All packets, which belong to a particular FEC and travel from a particular node, will follow the same path. In conventional IP forwarding, which is typical of the FEC establishment phase of MPLS, a particular router will typically consider two packets to be in the same FEC if there is some address prefix X in that router's routing tables such that X is the "longest match" for each packet's destination address. As the packet traverses the network, each hop in turn reexamines the packet and assigns it to a FEC.

In MPLS enabled networks, the assignment of a particular packet to a particular FEC is done just once, as the packet enters the network. The FEC to which the packet is assigned is encoded as a short fixed length value known as a "*label*". When a packet is forwarded to its next hop, the label is sent along with it; that is, the packets are "*labeled*" before they are forwarded. At subsequent hops, there is no further analysis of the packet's network layer header. Instead, the label is used as an index into a table, which specifies

the next hop, and a new label. The old label is replaced with the new label, and the packet is forwarded to its next hop. All subsequent forwarding are driven by the labels.

Fig. 2-1 shows the MPLS label format [12]. The MPLS header (called Shim header) is between layer 2 and layer 3 headers in an IP packet. It is 32 bits long. The label field (20 bits) carries the actual value of MPLS label.

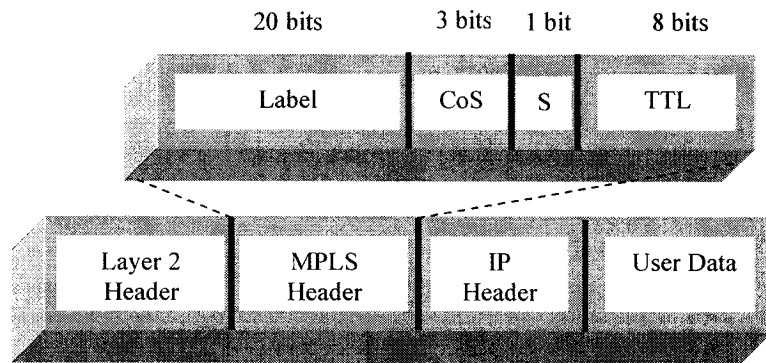


Fig. 2-1 MPLS header format

The CoS field (3 bits) is used to enforce certain quality of service and it can affect the queuing and discarding algorithms applied to the packet as it traverses through the network. The Stack (S) field (1 bit) is used to support a hierarchical label stack. Finally the Time-To-Live (TTL) field, which is 8 bits long provides conventional IP TTL functionality.

MPLS enables support of new features and applications as summarized in Table 2-1 [10-11,13-16].

### 2.2.2 Label Assignment Rules

In MPLS architecture, the decision to bind a particular label L to a particular FEC F is made by the LSR, which is downstream with respect to that *binding*. The downstream refers to the direction in which a user packet is sent. The *downstream* LSR then informs

the *upstream* LSR of the binding. Thus labels are "downstream-assigned", and are "distributed upstream", i.e. in the "downstream to upstream" direction [10, 13, and 18].

MPLS needs a mechanism for distributing labels in order to setup paths. Several protocols that can support label distribution are currently in operation. IETF developed a specific protocol to complement MPLS. It is called Label Distribution Protocol (LDP) [10,15-16 and 18].

Table 2-1 MPLS features

<i>Feature No.</i>	<i>MPLS Feature</i>
1	<i>MPLS supports streams and labels</i>
2	<i>Labels are local</i>
3	<i>MPLS can use various Layer 2 networks</i>
4	<i>MPLS uses edge device concept</i>
5	<i>MPLS is compatible with OSPF and BGP</i>
6	<i>QoS can be inferred from label</i>
7	<i>MPLS supports source (explicit) routing</i>
8	<i>MPLS supports traffic engineering (TE)</i>
9	<i>MPLS supports Virtual Private Network (VPN)</i>

### 2.2.3 Label Swapping

There are three types of MPLS routers as shown in Fig. 2-2 [10,15-16]:

- **Ingress LSR** (also called Label Edge Router- *LER*): Receives native mode user traffic (for example, IP datagrams) and classifies it into an FEC. It then generates MPLS header and assigns it a label. The datagram is encapsulated into the MPLS packet data unit (PDU), with MPLS header attached to the datagram. If it is

integrated with a specific QoS operation, the ingress LSR will condition the traffic in accordance with QoS requirement.

- **Transit (Interior) LSR:** Receives the packet and uses the MPLS header to make forwarding decisions. It will also perform label swapping. It is concerned with only the label header and not Layer 3 header.
- **Egress LSR** (also called Label Edge Router- *LER*): Performs the decapsulation operation and it removes the MPLS header from the datagram. Note that for certain type of flow, an LSR may play different roles depending on the location of customer premises equipment (CPE) and the location of LSR itself.

Fig. 2-2 shows the concepts behind label swapping in MPLS. Machines A, B, C and D are not configured with MPLS and are called customer premises equipment (*CPE*). Node E is the ingress LSR; nodes F, G, K and M are transit LSRs and node H is the egress LSR. For simplicity reasons, we have used generic address in this figure. For example the address for node C is " C ", which could be an IP address or some other address. LSR E receives an IP datagram from user node A on interface 1. This datagram is destined for node C. LSR E analyzes the FEC field, correlates the FEC with label 25, encapsulates the datagram behind a label header, and sends the packet to the output interface 3. The OUT entry in LSR E's table directs it to place label 25 onto the label header in the packet. This operation at LSR E is called *label push*.

After that, LSRs F and G process only the label header, and their swapping tables are used, for example at F, to swap label 25 for label 50, and LSR G to swap label 50 for label 7. Egress router H is configured to recognize label 7 on interface 6 as its own label; that is there is no more hops. Notice the OUT entry in H's table directs LSR H to send

this datagram packet to machine C on interface 4, which implies removing the label from this packet. This label removal is part of an operation called *label pop*.

Fig. 2-2 also shows the Label Switched Path (LSP), which is defined as the path through one or more LSRs, followed by packets in a particular FEC. The LSP in Fig. 2-2 spans the routers E, F, G and H.

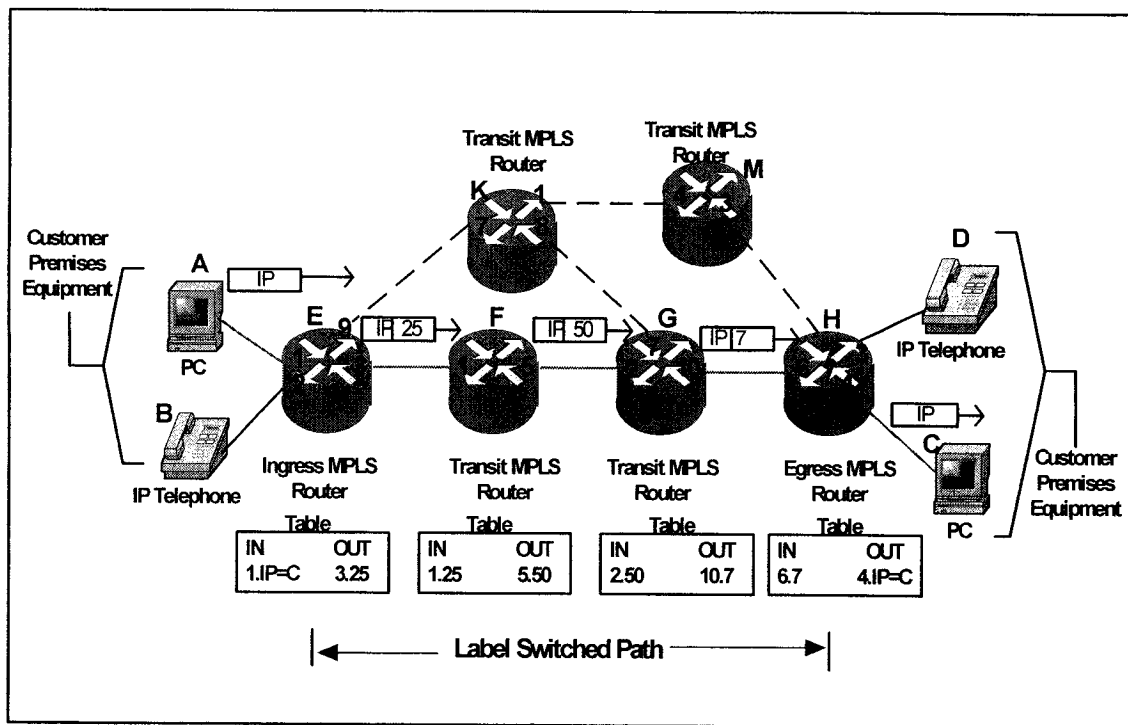


Fig. 2-2 Label swapping and forwarding.

## 2.3 MPLS Multicasting

Multicast routing proceeds by constructing IP multicast trees. The tree along which a particular multicast packet must get forwarded depends in general on the packet's source address and its destination address. Whenever a particular LSR is a node in a particular multicast tree, it binds a label to that tree. It then distributes that binding to its parent on

the multicast tree. (If the node in question is on a LAN, and has siblings on that LAN, it must also distribute the binding to its siblings. This allows the parent to use a single label value when multicasting to all children on the LAN.)

Different multicast protocols will therefore, in general, generate different trees. Several characteristics of these trees are discussed in the following subsections [2, 12, 19-22].

### **2.3.1 Aggregation and Granularity**

A key component of MPLS is that one or multiple flows may be assigned to the same label or flow. This is known as aggregation [15-16, 19-20]. Thus, a stream can range from fine to coarse granularity. The choice of label granularity balances the need to share the same label among many destinations with the need to maximize the switching benefits while preserving resources. Aggregation can reduce the number of labels needed to handle a particular set of packets and can also reduce the amount of label distribution control traffic needed. Given a set of FECs, which can be aggregated into a single FEC, it is possible to:

**a)** Aggregate them into a single FEC, **b)** aggregate them into a set of FECs or **c)** do not aggregate them at all.

### **2.3.2 Flooding and Pruning**

To establish a multicast tree, some IP multicast routing protocol (e.g. Distance Vector Multicast Routing Protocol (DVMRP) [2] or Protocol Independent Multicast- Dense Mode (PIM-DM)) [2] flood the network with multicast data [2, 19-30]. *Flooding* is the simplest algorithm that can be used to reach all members of a group. It is easy to implement and no routing tables have to be maintained for forwarding of the data.



However, it is extremely inefficient because the principle is similar to broadcasting. Flooding can place a high internal load on a network. In addition to that, *loops* may occur, which means that data can end up constantly circling the network. MPLS deals with this by using the TTL field, which indicates usually the maximum number of links that the data is allowed to pass. Flooding is suitable for large multicast groups. Fig. 2-3 (a) shows an example of packet flooding.

The branches can then be pruned by nodes, which do not want to receive the data of the specific multicast group as shown in Fig. 2-3 (b). This process is repeated periodically, thus generating a very volatile tree structure. Direct mapping of this dynamic L3 point to multi-point tree to L2 point to multi-point LSP is problematic because of limited label space, the signaling overhead and the setup time of the LSPs.

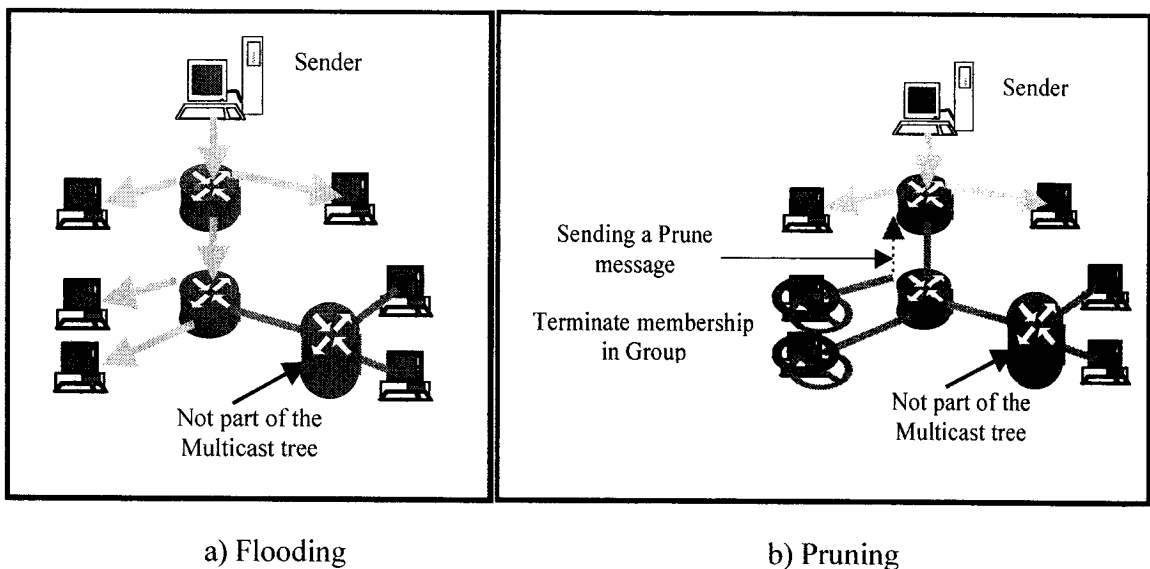


Fig. 2-3 Flooding and Pruning

### 2.3.3 Source and Shared Trees

As shown in Fig. 2-4, IP multicast routing protocols can create two types of trees [19-30]:

- Source tree (S, G), which is a tree per source (S) and per multicast group (G).
- Shared tree (\*, G), which is one tree per multicast group for all sources.

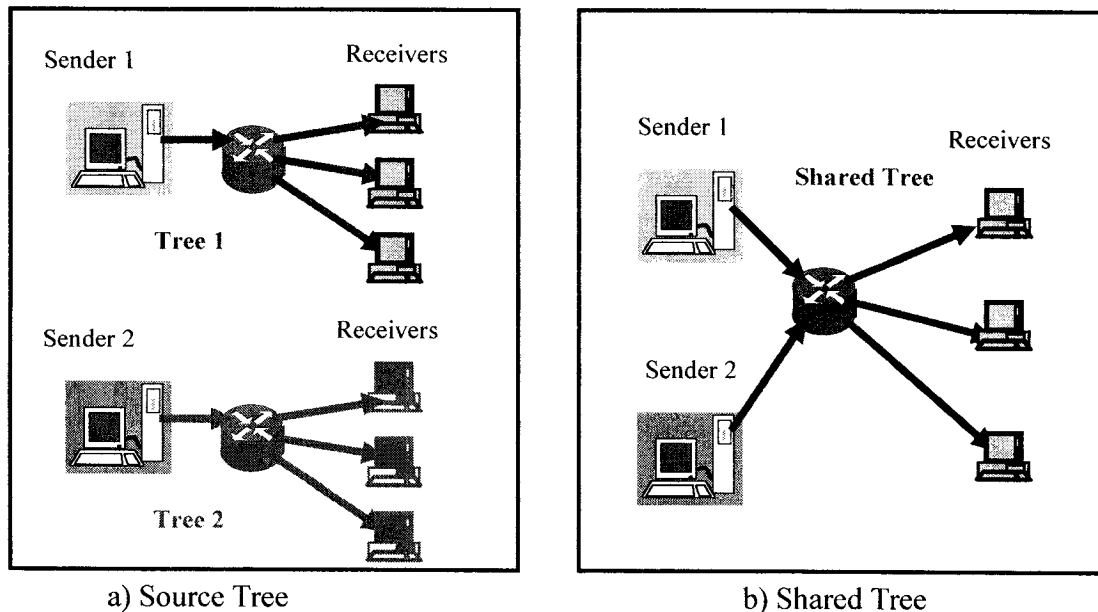


Fig. 2-4 Source and Shared Trees

The advantage of using shared trees, when MPLS and label switching is applied, is that shared trees consumes less labels than source trees (1 label per group versus 1 label per source and per group). Some protocols (e.g. Protocol Independent Multicast- Sparse Mode (PIM-SM) [2, 25-30]) support both source and shared trees. In addition to that, shared trees can be either unidirectional or bi-directional. For example, Core Based Tree (CBT) [2, 22] multicast routing protocol supports bi-directional shared trees.

### 2.3.4 Label Switched Patch (LSP) Triggers

The creation of an LSP for multicast streams can be triggered by three different events [15-16, 19-21]:

- 1- **Topology driven:** The L3 tree, which is available in the Multicast Routing Table (*MRT*), is mapped to an L2 tree. The mapping is done even if there is no traffic. The IP multicast routing protocol daemon maintains the MRT. The MPLS module maps this L3 tree topology information to L2 point to multi-point LSPs.
- 2- **Traffic driven:** Traffic driven triggers will only construct LSPs for trees that carry traffic. They consume fewer labels than the topology driven methods, as labels are only allocated when there is traffic on the multicast tree. However, this approach introduces a setup delay for the LSP.
- 3- **Request driven:** The control messages (e.g. multicast join/prune routing messages or resource reservation messages) trigger the setup of an LSP.

### **2.3.5 Label Advertisement**

In MPLS networks, label advertisement is required between peer LSRs. This can be achieved in two different ways [18-20]:

- 1- The use of a dedicated protocol like LDP protocol.
- 2- The use of piggybacking which enables us to carry the label advertisement message on the existing control messages rather than sending two separate messages.

### **2.3.6 Types of MPLS Routing**

MPLS provides two types of routing [19-21]:

- 1- Hop-by-hop routing where each LSR independently selects the next hop for a given FEC and label. The LSR uses any available routing protocols such as OSPF.

- 2- Explicit routing (similar to source routing) in which the ingress LSR specifies the list of nodes through which the packet should traverse. Resources may be reserved along the path to ensure QoS.

## **2.4 MPLS Multicast Considerations**

MPLS offers many advantages over IP, however there is a number of considerations and open issues [15-16, 19-22] that would be addressed in the following subsections.

### **2.4.1 Label Switched Path (LSP) Establishment Latency**

In MPLS, there will be some latency prior to a full establishment of LSPs [15-16,19-20]. This is due to label assignment messages overhead for traffic that would require a path to be put in place the moment the flow is detected. In this instance, the overhead will increase in relation to the number of flows being supported especially in the case of multicasting where a multicast tree should be constructed. In addition to that, when the label distribution is included as part of RSVP, the overhead and scalability of MPLS must be considered.

### **2.4.2 Limited Label Space**

MPLS can run on top of many L2 technologies such as Asynchronous Transfer Mode (ATM), Frame Relay (FR), and Ethernet, ... etc. ATM offers QoS and very high switching capabilities, but when used as L2 technology in the context of MPLS ATM or Frame Relay place several limitations. One of these limitations is limited label space. The number of bits available for a label can be small (e.g. Virtual Path Identifier/Virtual Circuit Identifier (VPI/VCI) space or Data Link Connection Identifier (DLCI) space),

limiting the number of LSPs that can be established. This is a major consideration when multicasting is used [19-20].

### **2.4.3 Security**

Security is an issue in MPLS networks, because the MPLS generic encapsulation inserts its header between the data link layer (L2) header and the network layer (L3) header as shown in Fig. 2-1. This may cause security procedures to fail, especially when some router may implement security procedures, which depend on the network layer header being in a fixed place relative to the data link layer header [19-20].

## **2.5 Quality of Service (QoS) in the Internet**

QoS is defined as the ability of a communication network to provide preferential treatment to some network traffic as apposed to all traffic being treated as best-effort or as soon as possible. QoS is a key network service criteria in the design and implementation of today's communication networks in order to meet the different applications requirements and to acquire different grades of network service in terms of bandwidth, packet delay, throughput, packet loss,...etc.

The QoS used in today's Internet is "best-effort" service or "as soon as possible" service, which is suitable for the traditional Internet applications such email, file transfer,...etc, because these applications can tolerate delay. However, this service is not suitable for demanding real time applications such as video conferencing, which need more bandwidth and low delay. QoS performance measures can be delay, throughput, delay jitter, packet loss and bandwidth. The main Internet traffic performance objectives

are maximization of bandwidth and throughput and minimization of delay, delay jitter and packet loss [31-41].

### **2.5.1 Integrated Services and Resource ReSerVation Protocol (RSVP)**

#### **Model**

Integrated services model (IntServ) is an IETF architecture [34] for providing QoS in the Internet. The concept is to reserve resources explicitly, bandwidth for example, for each individual flow to guarantee the required QoS. Applications must first set up paths and reserve resources before start transmission if they require a guaranteed service or a controlled-load service.

The Resource Reservation Protocol (RSVP) [34-38] is an IETF standard protocol. It is a signaling protocol that enables Internet applications to obtain special quality of service (QoS) for a data flow. RSVP is not a routing protocol; instead, it works in conjunction with routing protocols. RSVP occupies the place of a transport protocol in the OSI model. Routing protocols determine where the packet should be forwarded; RSVP is concerned with the QoS of the forwarded packet. RSVP is a receiver-based protocol; resource reservations requests are originated by the receivers of the service. This model is relatively complex and has difficulties in scaling to large backbones [31-33]. RSVP supports three traffic types: Best effort, controlled-load [37] and guaranteed services [38].

### **2.5.2 Differentiated Services (DiffServ) Model**

Because of the scalability problem of Integrated Services/RSVP, another scheme is introduced. The IETF DiffServ working group is looking at a more scalable model and

more likely to be easier to implement and deploy than IntServ/RSVP model [42-46]. The principle behind the DiffServ is to divide the traffic into many classes and treat them differently according to each class priority, especially when there is a shortage in network resources. The DiffServ is based on traffic aggregation rather than per-flow state and signaling at every hop as in IntServ/RSVP.

#### **2.5.2.1 DiffServ Architecture**

With DiffServ concept, a network is divided into domains. A distinction is made between DiffServ domains and domains that are unable to support DiffServ. In the DiffServ architecture, traffic that requires the same Per-Hop-Behavior (PHB) is aggregated into a single queue. As shown in Fig. 2-5, the DiffServ architecture focuses on the use of DiffServ (DS) byte, which is the redefined 8-bit Type of Service (TOS) field in the IPv4 header or the IPv6 Traffic Class octet as a QoS mechanism [44]. Applications can set the value of these fields according to its QoS requirements. Diffserv defines the layout for TOS field (DS field) and a basic set of rules for packet forwarding (Per-Hop Behavior, PHB). Packets are classified into the corresponding queues using their DiffServ Code Points (DSCP). Packets use DSCP bits in order to receive a particular PHB, or forwarding treatment. Marking, classification, traffic conditioning or policing are done at network boundaries (first router for example) and packet treatment and handling is carried on each network node [42-46]. Fig. 2-6 shows the logical operation of DiffServ classifier and scheduler at an intermediate router.

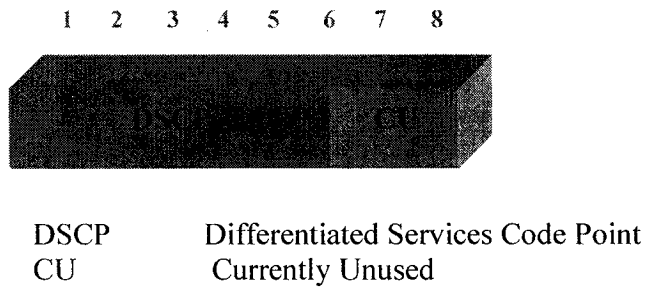


Fig. 2-5 Differentiated Services (DS) field structure

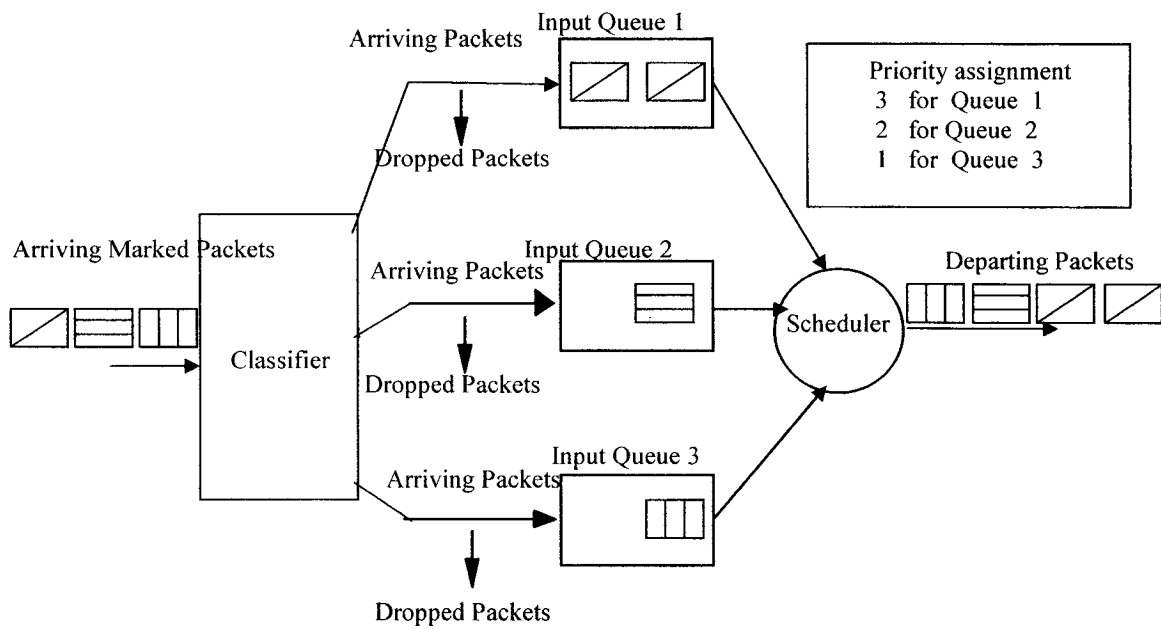


Fig. 2-6 Packet Classifier and Scheduler

We will illustrate some of the components that are shown in Fig. 2-6:

- 1- *Packet marking*: The ingress router sets the DS field of a packet as it enters the network to a particular codepoint, adding the marked packet to a particular DS behavior aggregate. The interior routers then can handle the marked packets differentially.



- 2- *Packet classification*: When a certain router receives a packet, it must check the DS field of the packet to determine if the packet should receive differential treatment, and then the classifier sends the packet to the appropriate queue.
- 3- *Packet queuing*: The router may use multiple queues.
- 4- *Traffic shaping*: Usually a traffic shaper has a finite-size buffer space. The shaper delays some or all of the packets in a traffic stream in order to bring the stream into compliance with the service level agreement. Because of insufficient buffer space to hold the delayed packets, packets may be discarded. In order to bring the traffic stream into compliance with the service level agreement, *dropper* discards some or all of the packets in a traffic stream. This process is known as "*traffic policing*" the stream. There are different mechanisms to drop packets in case of insufficient buffer space: Tail drop, head drop, random early discard (RED), random early discard with in and out (RIO).
- 5- *Packet scheduler*: The router may employ certain scheduling mechanism such that delay sensitive (for example real-time) traffic will service sooner. Since there are many scheduling algorithms, there are different types of queues such as : *First in first out (FIFO)* queues, *priority queues (PQ)*, *weighted round robin (WRR)* queues, and *weighted fair queues (WFQ)*.

The IETF DiffServ architecture provides three types of services: best effort service, which is the default service (DE), Assured service [47], where The assured-forwarding (AF) per hop behavior (PHB) is assigned to the assured traffic and Premium service [48] where the expedited-forwarding (EF) PHB is assigned to the premium traffic.

## **2.6 Reliable Multicast**

### **2.6.1 Introduction to Error Control**

The ability to detect errors when a transmission has been changed is called error detection and the ability to correct the detected error is called error correction. Error control coding provides the means to protect data from errors. Data transferred from one place to the other has to be transferred reliably.

Multicast has become an important component of today's Internet. In order to provide QoS in group communications for real time applications such as video conferencing, reliable multicasting is used. With reliable multicasting, all the receivers should receive all data packets correctly and in the right sequence. Therefore, error detection and recovery mechanisms are required in the implementation of reliable multicast. A number of efforts have been undertaken to provide reliability on top of IP multicast. Two error control strategies have been popular in practice. They are the **FEC (Forward Error Correction)** strategy, which uses error correction alone, and the **ARQ (Automatic Repeat Request)** strategy, which uses error detection combined with retransmission of corrupted data.

### **2.6.2 Error Control Strategies**

We will review in this section the two error control strategies ARQ and FEC [49-60]:

#### **2.6.2.1 Automatic Repeat Request (ARQ)**

ARQ is an error control mechanism, which can be accomplished by using error detection and retransmission. In ARQ strategy, when an error is detected at the receiver, a

request is transmitted to the sender to repeat the incorrect message, and this continues until the message is received correctly. ARQ is divided further into two subtypes:

1- Stop-and-wait ARQ, where the transmitter sends a code word to the receiver and waits for an acknowledgment from the receiver which would be either positive (ACK) or negative (NAK). If a positive ACK is received, which means that the receiver has received the code word correctly and no errors has occurred, the sender sends the next code word. However, if a negative ACK (NAK) is received which means the code word was received with errors, the sender resends the code that is in error. In case of noisy channel, the code word could be retransmitted many times before it is received correctly.

2- Continuous ARQ: It is also called sliding window protocol. With this type of ARQ, the sender transmits the code words to the receiver continuously (up to the maximum window size) and receives acknowledgements continuously. When a negative ACK (NAK) is received by the sender from the receiver, the sender begins the retransmission. There are two subtypes of the continuous ARQ strategy:

- Go-back-N ARQ: In go-back-N protocol, when a NAK is received by the sender, it resends that word plus all the words that follow it.
- Selective-repeat ARQ: In this case, when a NAK is received by the sender, it resends only those code words that are acknowledged negatively. Selective-repeat ARQ is more efficient than go-back-N in terms it consumes less bandwidth, however it requires more logic and buffering.

The error detection and recovery mechanisms must be extended to group communications. Mechanisms used for unicast communications cannot easily be

extended. A number of efforts have been undertaken to provide reliability in case of multicasting. The difficulty is with the scalability of traditional procedures to accommodate arbitrarily large, and thus heterogeneous, receiver sets. In a simple scenario, each receiver would be sending acknowledgment to the sender, which in case of large group could easily result in performance bottlenecks. This problem is referred to as *feedback implosion*. Feedback implosion can be avoided by keeping no state at the sender and making the receiver responsible for detecting loss. Such schemes are referred to as receiver-reliable, and often making it the receivers job to send NAKs.

There are two methods to send the repair packets to the receiver or group of receivers:

- 1- Multicast repairs: In case of receiving a NAK from one or more receivers the sender multicast again the repair packet to all receivers.
- 2- Unicast repairs: With unicast repairs, if the sender received a NAK from one or more receivers, it resends the repair packet to only the receivers who did not receive the packet correctly in a unicast manner.

The Multicast repairs method is simpler than the unicast repairs method and requires less overhead; however, the multicast repairs method consumes much more bandwidth.

#### **2.6.2.2 Forward Error Correction (FEC)**

FEC employs error-correcting codes that automatically correct errors detected at the receiver. Error control for a one-way system must be accomplished using FEC. One example is the magnetic tape storage system, where the information recorded on tape may be replayed some time later after it is being recorded.

The FEC strategy is mainly used in links where retransmission is impossible or impractical. The FEC strategy is usually implemented in the physical layer and is

transparent to upper layers of the protocol. When the FEC strategy is used, the transmitter sends redundant information along with the original bits and the receiver makes its best to find and correct errors.

Most of the FEC literature deals with error correction. That is, the ability to detect and repair bit-level corruption, as well as erasures (outright loss of data). Because the lower layers of the network will detect the corrupted packets and discard them, an IP or MPLS multicast application need not be concerned with corruption, but can focus on erasure correction only. The form of FEC utilized in the Multicast is known as  $(n,k)$  linear block encoding.  $k$  source packets are encoded into  $n > k$  packets, which constitutes an FEC group. Such that, any  $k$  of the encoded packets can be used to reconstruct the original  $k$  as show in Fig. 2-7.

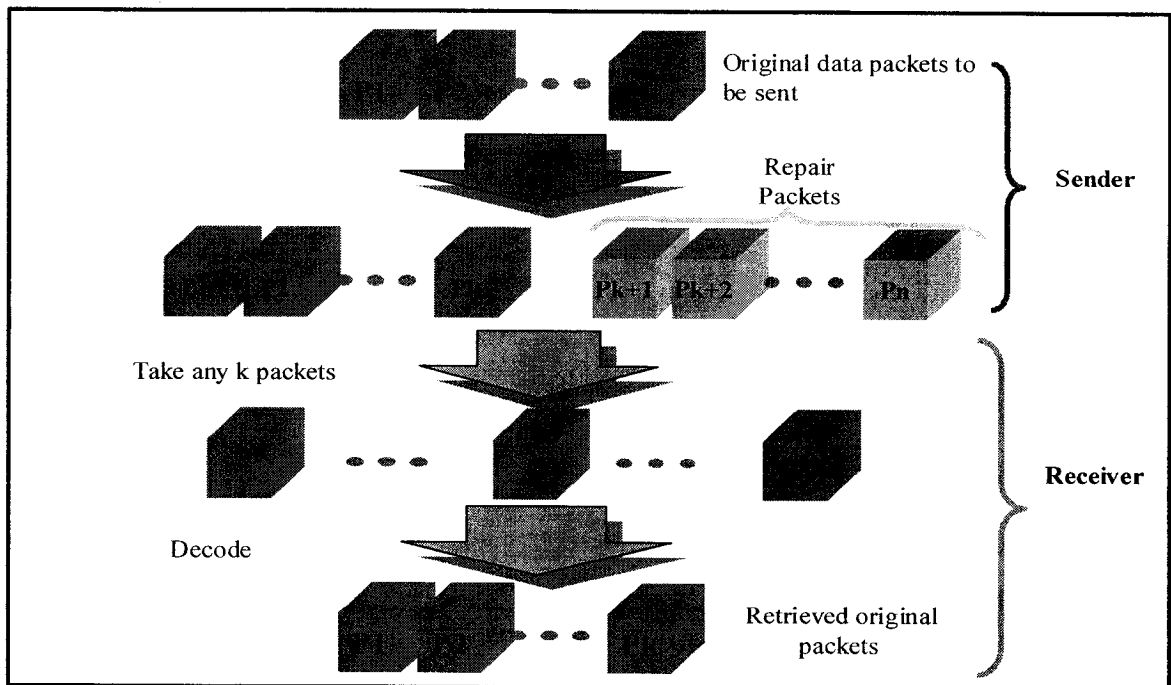


Fig. 2-7 FEC Operation

The major advantage of ARQ over FEC is that error detection requires much simpler decoding equipments than does error correction. In addition, ARQ is adaptive in the sense that information is retransmitted only when errors occur. However, when the channel error rate is high, retransmissions must be sent too frequently and the system throughput is lowered by ARQ due to bandwidth consumption.

### **2.6.2.3 Hybrid FEC/ARQ**

A hybrid FEC/ARQ strategy should be used where a combination of FEC for the most frequent error patterns, together with error detection and retransmission for the less likely error patterns is more efficient than ARQ alone. In this case, when FEC fails to correct errors at the receiver the receiver sends a NAK to the sender to retransmit the data in error. This hybrid FEC/ARQ strategy clearly carries the potential for improving throughput in two-way systems subject to a high channel error rate.

## **2.6.3 Reed-Solomon Codes**

### **2.6.3.1 Properties of Reed-Solomon Codes**

Reed-Solomon codes are block-based error correcting codes with a wide range of applications in digital communications and storage. Reed Solomon codes are a subset of Bose-Chaudhuri-Hocquenghem (BCH) codes and are linear block codes. A Reed-Solomon code is specified as  $RS(n,k)$  with  $s$ -bit symbols. This means that the encoder takes  $k$  data symbols of  $s$  bits each and adds parity symbols to make an  $n$  symbol code word. There are  $n-k$  parity symbols of  $s$  bits each. A Reed-Solomon decoder can correct

up to  $t$  symbols that contain errors in a code word, where  $n-k=2t$ . In order to be able to correct errors, Reed-Solomon codes require a minimum distance of:

$$d_{min}=2t + 1 = n-k + 1$$

A typical Reed-Solomon code word is shown in Fig. 2-8 (it is called a systematic code because the data is left unchanged and the parity symbols are appended):

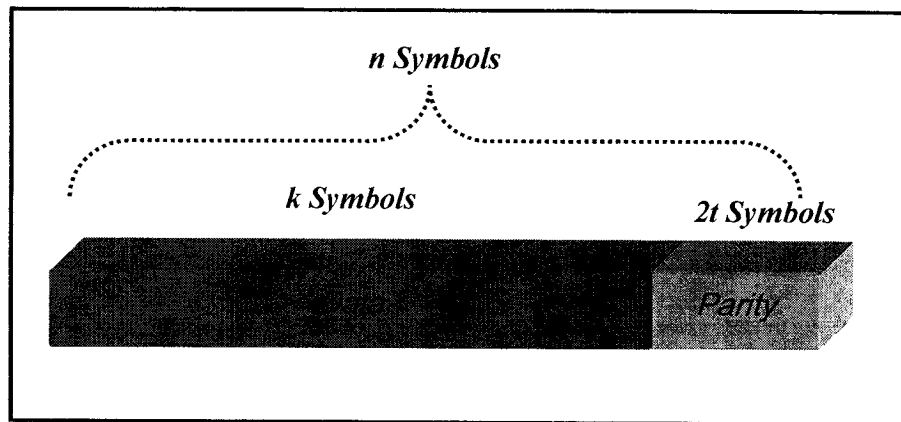


Fig. 2-8 Reed-Solomon Code Word

Coding rate is defined as  $r = k/n$ .

A very well known Reed-Solomon code is RS(255,223) with 8-bits symbols. Every code word contains 255 code word bytes, of which 223 bytes are data and 32 bytes are parity.

For the RS(255,223) code:  $n = 255$ ,  $k = 223$ ,  $s = 8$ ,  $r = 223/255$ ,  $2t = 32$  and  $t = 16$ .

Given a symbol size  $s$ , the maximum code word length ( $n$ ) for a Reed-Solomon code is  $n = 2^s - 1$ . For example, the maximum length of a code word with 8-bit symbols ( $s=8$ ) is 255 bytes.

### 2.6.3.2 Reed-Solomon Decoding

Reed-Solomon algebraic decoding procedures can correct errors and erasures. An erasure occurs when the position of an erred symbol is known. A decoder can correct up

to  $t$  errors or up to  $2t$  erasures. Erasure information can often be supplied by the demodulator in a digital communication system, i.e. the demodulator "flags" received symbols that are likely to contain errors. One of three possible outcomes could occur when a code word is decoded:

If  $2s + R < 2t$  ( $s$  errors,  $R$  erasures) then the original transmitted code word will always be recovered, ELSE The decoder will detect that it cannot recover the original code word and indicate this fact.

OR The decoder will mis-decode and recover an incorrect code word without any indication.

The probability of each of the three possibilities depends on the particular Reed-Solomon code and on the number and distribution of errors.

## **2.7 Conclusions**

An overview of the literature background was presented in this chapter. This overview has introduced the new promising protocol MPLS. In addition to that, MPLS multicast considerations were presented and explained. The definition of QoS in the Internet and how it can be achieved (using mainly IntServ and DiffServ) was also described. Finally, reliable multicast and the two main error control strategies (FEC and ARQ) were presented.



## **CHAPTER 3 RELATED WORK OF MULTICASTING TECHNOLOGIES**

### **3.1 Introduction**

Due to the QoS-aware group applications such as video/voice conferencing, streaming audio/video, software upgrading and database updates, etc, there is a great demand for a more efficient Internet that could support QoS multicasting. On the other hand, reliability became an important issue in the area of multicasting to guarantee the delivery of data and to achieve the required QoS level. Different multicast technologies have been proposed and some are currently in use to provide group communications in the Internet. In addition to that, QoS in heterogeneous networks needs an extensive study since there are few papers that addressed this problem. In this chapter, a survey-like of multicasting technologies that are related to thesis work will be summarized. This chapter will address the following subjects: congestion control, reliable IP multicast, MPLS multicast, DiffServ Multicast, DiffServ/MPLS multicast and finally QoS in heterogeneous networks.

### **3.2 Congestion Control and Traffic Policing**

In the Internet and especially MPLS based networks, most traffic sources are bursty. Such a bursty traffic source will not require continuous allocation of bandwidth at its peak rate. Statistical multiplexing can be used to gain bandwidth efficiency, allowing more traffic sources to share the bandwidth. However, if a large number of traffic sources become temporarily active simultaneously, severe network congestion can result.

Therefore, to prevent this situation and especially in case of multicasting, there should be some congestion control schemes [15, 61-65].

Call admission control is not sufficient to prevent congestion, mainly because users may not stay within the connection parameters according to the Service Level Agreement (SLA) between the user and the Internet Service Provider (ISP). Therefore, after a connection is set up, some flow control is still required to provide good performance and guaranteed quality of service among the users. This kind of control is based on the declared parameters, and needs a policing procedure to ensure that any change in the user's traffic characteristics will not affect the overall performance of the network. Therefore, the purpose of the policing mechanism is to avoid short-term congestion caused by bursts of packet transmissions [15, 61-65].

When a source exceeds its negotiated parameters, the network could take either one of mainly three actions: Packet discarding, packet buffering or violation tagging.

The Token bucket or leaky bucket [61-65] method is one of the typical bandwidth or traffic enforcement mechanisms used in MPLS networks; this method can enforce the peak or average bandwidth and the burst factor of a traffic source. The Token bucket scheme was first introduced in [63]. Since then a number of its variants have been proposed. The basic idea behind this approach is that a packet, before entering the network, must obtain a token from the token pool. An arriving packet will consume one token and immediately depart from the token bucket if there is at least one token available in the token pool. Tokens are generated at a constant rate and placed in a token pool. There is an upper bound on the number of tokens that can be waiting in the pool (bucket size  $M$ ) and tokens arriving at a time when the token pool is full are discarded.

The size of the token pool imposes an upper bound on the burst length and determines the number of packets that can be transmitted back to back, thus controlling the burst length. The token bucket scheme is illustrated in Fig. 3-1.

The Exponentially Weighted Moving Average (EWMA) is another used policing mechanism in MPLS networks. It uses fixed consecutive-time windows, i.e. the window size  $T$  is constant and a new window is triggered immediately after the preceding window ends. The maximum number of accepted packets varies from one window to the next [64], as shown in Fig. 3-2.

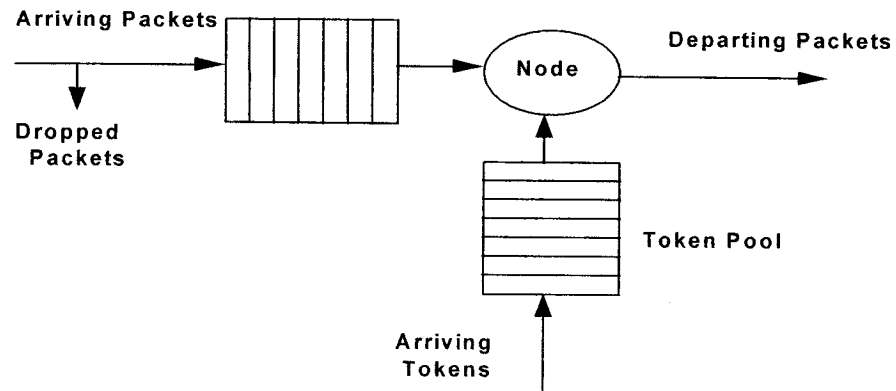


Fig. 3-1 Token Bucket Scheme

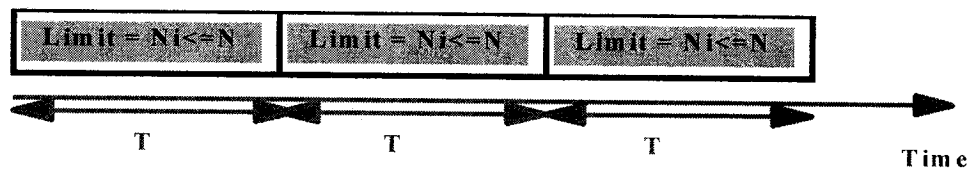


Fig. 3-2 Exponentially Weighted Moving Average Scheme

In particular, the maximum number of packets in the  $i$ th window ( $N_i$ ) is a function of the allowed mean number of packets per interval  $N$  and exponentially weighted sum of the number of accepted packets in the preceding intervals ( $X_i$ ) according to the rule:

$$N_i = \frac{N - S_{i-1}\gamma}{1 - \gamma} \quad \text{Where } 0 \leq \gamma < 1 \text{ and } S_{i-1} = (1 - \gamma)X_{i-1} + \gamma S_{i-2} \quad (3-1)$$

With  $S_0$ , being the initial value of the EWMA. The factor  $\gamma$  controls the flexibility of the algorithm with respect to the burstiness of the traffic. If  $\gamma=0$ ,  $N_i$  is constant and is always equal to  $N$ . A packet that pushes the average rate over a predefined average rate is nonconforming. So, for a packet arriving at time  $t$ :

If  $N_i > N$  then this packet is nonconforming and is to be discarded

else it is a conforming packet

Note that in Token Bucket scheme [61-65] or EWMA [64] the length of the bucket size and the window size directly affects the source throughput. That is because, when the bucket size is full or when the old window expires, both schemes will lose tokens (or credits) which are considered as lost chances for a packet to leave the current node. [61-65] did not consider what is the proper bucket size or the proper window size.

### 3.3 Reliable IP Multicast

A number of books, RFCs and research papers have addressed extensively the IP multicast issue [1-9]. In addition to that, a number of books and research papers have addressed the reliable multicast transport protocols [1,25-30]. " Different applications have different requirements of a reliable multicast protocol, and these requirements constrain the design space in ways that two applications with differing requirements often cannot share a single solution "[49]. There are many ways to provide reliability for transmission protocols and to ensure the correct delivery of data. In [49-60], error control strategies were described and used in order to achieve reliable multicasting.

In [54], an algorithm to estimate the optimal number of initial parity packets with prior knowledge of neither the population size of the multicast group nor the transmission conditions inside the network is proposed. A proactive integrated FEC/ARQ protocol that uses the mentioned algorithm is described, and the performance of this technique is studied. [55] describes the design and implementation of a system that provides reliable multicasting based on FEC and ARQ requests. In [56], a scalable reliable multicast (SRM) framework for light-weight sessions and application level framing has been described. The work in [56] has focused on SRM's request and repair algorithms for reliable delivery of data. However, it did propose a complete set of algorithms for implementing local recovery. In SRM [56], for NAK suppression, a receiver waits for a random time before sending a NAK, and refrains from sending a NAK if it receives a NAK from another receiver for the same packet. However, this mechanism may operate poorly when the loss occurs at the source link.

[57] determines and compares the maximum throughputs of the sending and receiving hosts for generic sender-initiated (A) and receiver-initiated NAK (N1) protocols. [67] proposes and analyzes the delay of three-reliable multicast protocols namely sender-initiated, receiver initiated and second receiver-initiated protocols.

In [59], a framework is developed which allows one to model analytically the impact of FEC on the average number of transmissions necessary to transmit a packet to all members of the multicast group. Different multicast tree topologies and different multicast group sizes are examined. One of the findings is that the shared part of the multicast tree is not always the best part to employ FEC.

In [60], a repair technique that combines FEC with ARQ is presented. The beauty of the proposed technique is its ability to reduce delay in reliable multicast delivery by sending repairs proactively (i.e. before they are required).

[66] proposes and describes a new multicast traffic performance analyzer considering routing protocols DVRMP and PIM-SM. In addition to that, it reports the results of analyzing multicast datagrams transmitted over the Internet.

[67] considers reliability through the use of ACK (when a packet is received) or NAK (when a packet is not received) only. In [67], expressions for the overall delay are derived for all three protocols: sender-initiated (A), receiver-initiated NAK protocol (N1) where NAKs are returned to the sender via point-to-point channel and receiver-initiated NAK protocol (N2) where NAKs are multicast to the sender by receivers.

The overall delay expression for receiver-initiated (N1) QoS protocol is given by [67] as:

$$E[S_{N1}] = (E[W_{N1}^S] + E[W_{N1}^R] + 2E[X] + \tau) + \frac{p^2(T_R + E[W_{N1}^R] + E[Y])}{(1-p)} + p(E[D_{N1} + \tau]) \quad (3-2)$$

Where  $E[W_{N1}^S]$  is mean waiting time at the sender and is given by:

$$E[W_{N1}^S] = \frac{\lambda_t^S E[X^2] + \lambda_r^S (E[X^2] + E[Y^2] + 2E[X]E[Y]) + \lambda_n^S E[Y^2]}{2(1 - \rho_{N1}^S)} \quad (3-3)$$

In which, the sender processes three flows. The first corresponds to the original transmissions. The second corresponds to the arrival of NAKs that trigger the retransmission and the third correspond to the arrival of NAKs that are processed but do not generate a retransmission. The corresponding flow rates are:

$$\lambda_t^S = \lambda \quad \{ \text{Poisson arrivals assumed and t denotes original transmission} \}.$$

$\lambda_r^S = \lambda(E[M] - 1)$  {  $E[M-1]$  = average number of times a packet is transmitted, and  $r$  denotes retransmissions }

$\lambda_n^S = \lambda(R E[M^{(r)}] - E[M] - R + 1)$  {  $M^{(r)}$  is the number of times a packet must be transmitted before a receiver receives it correctly and  $R$  is the number of receivers. }

The service times for the previously mentioned flows are :  $X$ ,  $X+Y$  and  $Y$  respectively.

The load at the sender  $\rho_{N1}^S$  is given as:

$$\rho_{N1}^S = \lambda(E[X]E[M]) + R(E[M^{(r)}] - 1)E[Y] \quad (3-4)$$

There are two work flows through the receiver under N1 protocol: one corresponds to the arrival of data packets from the sender and the other corresponds to the self-generated NAK by the receiver. These rates are:

$$\lambda_t^R = \lambda E[M](1 - p) \quad \text{and} \quad \lambda_n^R = \lambda(E[M^{(r)}] - 1) \quad \text{where } p \text{ is the packet loss probability. Note the respective service times are } X \text{ and } Y.$$

Hence, the load at the receiver:

$$\rho_{N1}^R = \lambda(E[X]E[M](1 - p) + (E[M^{(r)}] - 1)E[Y]) \quad (3-5)$$

Therefore, the mean waiting time at a receiver is given by:

$$E[W_{N1}^R] = \frac{\lambda_t^R E[X^2] + \lambda_n^R E[Y^2]}{2(1 - \rho_{N1}^R)} \quad (3-6)$$

It is assumed that all participants in the multicast are separated from each other by a delay of  $\tau$ .  $T_R$  is the length of the time out period at the receiver before detecting a loss.  $E[D_{N1}]$  is the average length of the loss detection phase which ends after sending a NAK to the sender.

In [68], a queuing analysis of a simple FEC scheme for interactive IP telephony is carried out. FEC was found not scale well and the audio quality will deteriorate for any amount of FEC and for any offset (the redundant information that are added to the original information). However, [68] did not consider the multicast issue.

Active Parity Encoding Services (APES) [69], achieves efficiency in terms of network bandwidth due to additional network support called repair servers (RS) which is another reactive FEC approach. APES sends parity packets in place of retransmissions. Receivers reconstruct original data packets from received packets. RS ensures that each of its downstream receiver get at least  $k$  distinct packets.

In [70], a hierarchy of RSs, where each RS works for receivers or RSs in its repair domain is presented. In a hierarchy of RSs, according to loss conditions of each repair domain, each RS decides sufficient amount of redundancy of FEC. Applying FEC to each repair domain independently, APES can achieve efficient bandwidth utilization. In order to achieve this, many RSs are needed which add more packet processing costs at RSs or receivers. The three protocols proposed by [69] are:

- 1- Store-Data-Build-Repairs Protocol (SDBR): Once a repair server reliably obtains  $k$  source packets, it reproduces (via FEC encoding) the  $k$  original data packets, which it subsequently buffers.
- 2- Build-Repairs-Store-Repairs Protocol (BRSR): A repair server decides in advance on a fixed number,  $b$ , of repairs per block to generate via FEC encoder. Here, the repair server does not buffer the source packets, but merely supply them as they arrive to the FEC encoder. This method is called *on-the-fly encoding*.



- 3- The Get-Repairs-Store-Repairs Protocol (GRSR): The repair server does not require FEC encoding capability. Instead, it requests  $b$  repair packets from the sender, which it buffers. Once it obtains the  $b$  packets, it behaves identically to BRSR.

The probability of losing exactly  $j$  out of  $k$  packets is given on [69] as:

$$\gamma_j^k(p) = \binom{k}{j} p^j (1-p)^{k-j} \quad (3-7)$$

It is assumed that when a receiver losses  $m$  of  $k$  packets in a block requires the repair server to reliably transmit packets  $k+1$  to  $k+m$ .

The bandwidth computations **from repair server to receivers** for the three protocols are [69]:

$$E[T_{\text{SDBR}}] = \sum_{j=0}^{\infty} 1 - [1 - \sum_{m=0}^{k-1} \gamma_m^{k+j}(1-p)]^r \quad (3-8)$$

$r$  is the number of downstream receivers for the repair server.

Let  $\tau_i$  be a random variable that equals the number of times that packet  $i$  is transmitted. For  $i \leq k$ , we have  $E[\tau_i] = 1$ , since the packet is transmitted at most once. For  $k < i \leq 2k$ , a packet transmitted as many times as needed by some receivers.

Therefore,

$$E[T_i] \leq \sum_{j=0}^{\infty} q_i(j) \quad (3-9)$$

$$\text{and hence, } E[T_{\text{BRSE}}] = E[T_{\text{GRSR}}] \leq k + \sum_{i=k+1}^{2k} E[\tau_i] \quad (3-10)$$

It is found in [69], for reasonable loss rates, BRSR and GRSR do not use substantially more bandwidth than SDBR between the repair server and the receivers.

The bandwidth computations **from sender to repair server** [69]:

$$E[A] = \sum_{i=k+b+1}^{2k} E[\tau_i] \quad (3-11)$$

Where  $A$  is a random variable equal to the number of additional transmissions the sender must make to a repair server.

[71] proposes local FEC, where FEC is applied to source link and receivers do not have to support FEC encoding/decoding. Receivers just have to operate NAK based mechanisms and no additional operations required. In [71], a procedure very similar to [67] is followed to calculate the overall delay. Let  $T$  denotes the overall delay, the expected value of  $T$  is given by:

$$E[T] = (1 - p_1) \{E[D_{GR}] + E[W_R] + E[X] + (\tau - \tau_{GR})\} + p_1 (E[D] + E[U] + E[B]) \quad (3-12)$$

Where  $p_1$  is the original packet loss probability between the sender and the receiver and is given by:

$$p_1 = 1 - (1 - p_{GR})(1 - p_d) \quad (3-13)$$

$p_{GR}$  is the probability that gateway router (GR) cannot recover data packet  $i$ .  $E[D_{GR}]$  is the mean time from the initial arrival of a packet at the sender to the time of forwarding at the GR.  $E[W_R]$  is the mean waiting time at the receiver.  $E[X]$  is the mean service time of a data packet or a parity packet.  $E[D]$  is the mean length of the loss detection phase.  $[\tau - \tau_{GR}]$  is the propagation delay between a receiver and the GR.  $E[B]$  and  $E[U]$  represent the mean lengths of the random delay phase and the loss recovery phase respectively.

[72] examined an approach for providing reliable, scalable multicast communications by using multiple multicast channels for recovery of lost packets. In the approach, rather than having all receivers receive all retransmitted packets, the multiple multicast channels are used to allow only these receivers that actually want a particular packet to actually receive that packet. In [72], infinite number of multicast channels and a limited number

of multicast channels scenarios are considered. It also considers two types of sender behavior: the one to many and many to many scenarios.

However, one can notice that none of these previously mentioned papers in the current section [49-60] and [66-72] has addressed the adoption of DiffServ into reliable IP multicast networks.

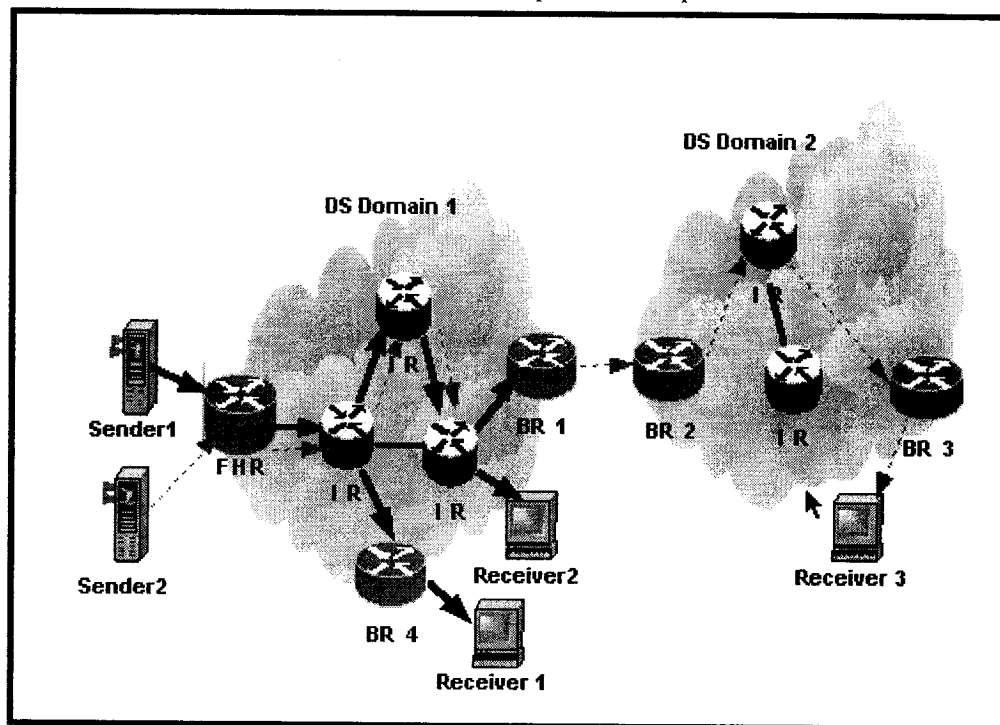
### **3.4 DiffServ Multicasting**

The current Internet infrastructure only supports best effort service, which is inadequate for QoS-sensitive application such as audio/video conferencing, distance learning, etc. One of the promising architectures to provide QoS in the Internet is DiffServ. Expedited Forwarding (EF) and Assured Forwarding (AF) are the two main QoS types defined in DiffServ architecture.

The DiffServ multicasting would be a useful application because DiffServ provides a method of service differentiation, which is needed in the next generation Internet. The area of DiffServ multicasting has received relatively little attention until recently. A few number of papers have addressed the DiffServ multicast issue [73-82]. Using DiffServ to provide QoS multicast poses several challenges:

- 1- The multicast tree changes dynamically due to the frequent change of the number of receivers, which is, know as join/leave problem.
- 2- Heterogeneous receiver resource requirements of multicast since each receiver may have its own resource requirement, which cannot be predicted in advance.
- 3- When internetworking different DiffServ domains that have different SLAs, mapping between different SLAs is needed.

Another well-known problem that could happen in DiffServ multicasting and it should be avoided under any circumstances is called the Neglected Reservation Subtree Problem (NRS Problem) [73, 74]. This problem could happen when a new receiver joins an IP Multicast group, a new subtree is added, which connects the new receiver to the already existing multicast tree. Because of tree expansion and missing per-flow classification mechanisms, the new receiver will implicitly use the service of better quality. If the additional amount of resources which are consumed by the new part of the multicast tree are not taken into account by the domain management, the currently provided level of quality of service of other receivers (with correct reservations) will be adversely affected or violated. This negative effect on existing traffic contracts by a neglected reservation. Fig. 3-3 shows an example of NRS problem.



FHR First Hop Router  
 IR Internal Router  
 BR Border Router

Fig. 3-3 NRS Problem in DiffServ multicast

At the beginning, there are two initial multicast groups. Group 1, which consists of sender 1 and receivers 1 and 2, and group 2 which consists of sender 2 and receiver 3. Assume that group 1 requires 30% of the bandwidth to provide Expedited Forward (EF) QoS to its members while group 2 reserves 20% of the bandwidth to provide EF QoS to its members. Up to now, there is no appearance of the NRS problem. The problem could arise when receiver 3 joins the multicast group 1. Since receiver 3 is in DS domain 2, which is connected to the DS domain 1 using Border router BR1. Since BR1 is also an egress router, which is equipped with a traffic policing function. BR1 knows that DS domain 2 uses 30% extra bandwidth without permission after receiver 3 joined group 1. Consequently, the policing component in the egress border router (BR1) drops packets until the traffic aggregate is in accordance to the traffic contract (20% of bandwidth). However, during packets dropping, the router cannot identify the responsible flow (because of missing flow classification functionality), and, thus randomly discards packets, whether they belong to a correctly reserved flow or not. As a result, there will be no longer any service guarantee for the reserved flows. This is called the NRS problem. Other types of NRS problems and solutions can be found in [73, 74].

There are different ways to support multicast in DiffServ networks. Core routers can be multicast-capable, multicast-incapable or multicast traffic is encapsulated to transfer to core routers. If core routers support multicast they have to maintain a multicast tree state per group. This makes core routers complex and not scalable, as they have to check the state for every multicast tree. If core routers are unaware of multicast traffic, traffic is replicated at boundary routers. Striegel and Manimaran [75] have given a new architecture for multicast support in DiffServ domain. In their DiffServ Multicast

(DSMCast) [75] approach, core routers do not need to maintain multicast tree. The tree information is encapsulated in packet's header, and leaves core routers a little simpler and more scalable. However, encapsulation incurs additional costs. "Fat" header consumes additional bandwidth for each packet, and additional CPU cost is also incurred due to header processing. Under basic DSMCast model [75], the tree is built based solely on the network topology. Multicast traffic is transmitted like unicast apart from that core routers have to inspect the header to make replication when needed. The extension headers include identification field for DS core nodes, appropriate branching information, tunneling bit to bypass DS-non-capable nodes, and adaptive DS field to adapt to the heterogeneous DSCP requirements by the different receivers.

The request that receivers join/leave the group is forwarded by the egress router to the ingress router to be processed. The construction of the multicast tree is then done by the ingress router. Member's join/leave structure is discussed in [76].

Bless and Wehrle [77] add an extension of DS entry to multicast routing table to support heterogeneous DiffServ multicast group. This makes different branches in the same multicast group to get different QoS possible, but routers have to maintain a relatively larger routing table.

One can notice that reliable DiffServ multicast is an open research area. To the best of our knowledge, none of current DiffServ multicast papers [73-82] has addressed the adoption of reliable multicast using FEC/ARQ.

### **3.5 MPLS Multicasting**

MPLS multicasting and MPLS multicast considerations were introduced in sections 2.3 and 2.4 respectively. MPLS multicast is still an open issue [19]. Recently, MPLS has

received a great deal of researchers' attention. A number of papers have addressed this issue [19-24] and [83-91]. These papers have concentrated mainly on three areas within the context of MPLS multicast, namely:

- 1- MPLS multicast architecture proposal, which mainly concentrates on the description of MPLS multicast and how it works.
- 2- MPLS routing (and rerouting) problems, which provide descriptions of such problems and suggest solutions.
- 3- MPLS label aggregation

Most of these papers were either descriptive or use simulation only. One can notice that reliable MPLS multicast is an open research area and to the best of our knowledge, none of current MPLS multicast papers [19-24] and [83-91] has addressed the adoption of reliable multicast using FEC/ARQ.

### **3.6 DiffServ/MPLS Multicasting**

The combined use of DiffServ and MPLS technologies is a promising way to provide QoS in the Internet, while effectively using network resources [92]. In addition to that, this combination will provide network reliability and adaptation of node and link failures.

However, there is a difference between DiffServ and MPLS. DiffServ is a layer 3 service while MPLS combines the flexibility of layer 3 routing and layer 2 switching(between layers 2 and 3). There are two basic problems for MPLS support of DiffServ:

- 1- The DiffServ DSCP has 6 bits whereas MPLS has 3 experimental bits (EXP) (or CoS field) as in Fig. 2-1.

2- The DSCP is carried in the IP header, where LSRs examine only the label header.

To carry DiffServ traffic over an MPLS network efficiently, a mapping between DiffServ classes and LSPs is needed. The solutions to the previously mentioned problems are given in [92].

There are two solutions defined: (1) EXP Inferred-PSC (PHB Scheduling Class) LSP (E-LSP), and (2) Label-Only-Inferred-PSC LSP (L-LSP).

### **1- EXP-Inferred-PSC (PHB Scheduling Class) LSP (E-LSP)**

E-LSP determines the PHB of a packet solely from the EXP field, and thus can support up to only 8 PHBs per E-LSP. The EXP field conveys the queuing, scheduling, and drop precedence to the LSP. PHB signaling can be used to explicitly signal the supported PHBs during LSP setup, but is not required (i.e. pre-configured PHBs).

### **2- Label-Only-Inferred-PSC LSP (L-LSP)**

Packets in a micro flow must maintain the same order from the ingress LSP to the egress LSP, so they belong to the same PHB Scheduling Class (PSC) [88], which is a PHB group such that the order of packets in the group must be preserved, and are placed in a common queue. The set of BAs whose order must be maintained during transmission constitutes an Ordered Aggregate (OA).

L-LSP determines the PHB of a packet from both the Label and EXP fields. The Label field determines the PSC (queuing and scheduling) while the EXP field determines the PHB (drop precedence). An arbitrarily large number of PHBs can be supported. The DiffServ object defined in the Resource Reservation Protocol (RSVP) extension or the DiffServ TLV defined in the Label Distribution Protocol (LDP) extension can be used to



support PHB scheduling group signaling, which is used to signal the PSC during L-LSP establishment [92].

In [93], an architecture called aggregated QoS Multicast (AQoS M) to provide QoS multicast support is proposed. Using same concepts of aggregate multicast used in [84], AQoS M can support QoS multicast scalably and efficiently in DiffServ supported MPLS networks. Aggregate multicast [84] is a scheme proposed to reduce multicast states. The idea is to force the multicast groups to share a single distribution tree. The enforcement takes place at the border routers of the network. Data packets from different groups are multiplexed on the same distribution tree, called **aggregated tree**. In [93], a network is modeled as an undirected graph  $G(V,E)$ . Each edge  $(i,j)$  is assigned a positive cost  $c_{ij}=c_{ji}$  which represents the cost to transport a unit of data from node  $i$  to node  $j$  or the reverses. Given a multicast tree  $T$ , the total cost to distribute a unit of data over this tree is:

$$C(T) \approx \sum_{(i,j) \in T} C_{ij} \quad (3-14)$$

If every link is assumed to have equal cost 1, tree cost is simply  $C(T)=|T| - 1$ , where  $|T|$  denotes the number of nodes in  $T$ . A “ native” multicast tree (e.g. using PIM-SM denoted by  $A$ ), which satisfies the membership and QoS requirement of a multicast group  $g$  is denoted by  $T_A(g)$ , while  $T(g)$  defines the aggregate tree which  $g$  uses to transmit data.

It is possible that  $T(g)$  does not have a perfect match with group  $g$ , which means that some of the leaf nodes of  $T(g)$  are not the member nodes of  $g$ . Then packets reach some destinations that are not interested on receiving them. Hence, there is a bandwidth overhead. Assume an aggregate tree  $T_0$  is used by groups  $g_i$ ,  $1 \leq i \leq n$ , each of which has a native tree  $T_A(g_i)$ , then the average percentage bandwidth overhead for  $T_0$  is given by [93] as:

$$\delta_A(T_0) = \frac{\sum_{i=1}^n B(g_i) \{C(T_0) - C(T_A(g_i))\}}{\sum_{i=1}^n B(g_i) C(T_A(g_i))} \quad (3-15)$$

Where  $B(g)$  is the bandwidth requirement for group  $g$ . [93] defines and uses 4 performance metrics to quantify the performance of AQoSM using simulations and these metrics are:

1- Number of MPLS trees.

2-Number of label forwarding entries installed in all routers

3- Request rejection ratio, which is defined as:

$$RR_{ratio} = \frac{N_R(t)}{N_A(t)} \quad (3-16)$$

Where  $N_A(t)$  denotes the number of group requests arriving in time period  $t$  after the steady state is reached and  $N_R(t)$  denotes the number of group requests which are rejected.

4- Tree setup ratio which is defined as:

$$TS_{ratio} = \frac{N_A(t) - N_M(t) - N_R(t)}{N_A(t)} \quad (3-17)$$

Where  $N_M(t)$  denotes the number of group requests which can be matched to some existing tree.

An attempt to explain the concepts of DiffServ + MPLS and illustrating its effectiveness by performing a simulation using Network Simulator (ns-2) is carried in [99]. The results of [99] show the fast rerouting feature of MPLS and how it alleviates the problem of link failures in DiffServ networks.

A network performance optimization problem related to traffic engineering over MPLS is considered in [100], where a dynamic traffic engineering and assignment

methodology to adaptively map ingress traffic into several parallel LSPs in MPLS network. Within the proposed framework, a set of parallel disjoint LSPs is modeled by parallel queues and a partitioning algorithm is devised for different service classes. However, [100] did not consider the multicast issue.

A number of research papers have addressed the adoption of DiffServ with MPLS [92-100]. However, only unicast operation is defined in [92], while multicast communications require further study. In addition to that, reliable DiffServ/MPLS multicast is an open research area and none of current DiffServ/MPLS papers [92-100] has addressed the adoption of reliable multicast using FEC/ARQ.

### **3.7 QoS in Heterogeneous Networks**

A number of papers have addressed the QoS issue in heterogeneous networks [101-112]. Many of them concentrated on the wireless heterogeneous IP networks [101-106]. In [107], topology discovery in heterogeneous IP networks was conducted. Multicast issue in heterogeneous networks was studied by [108-112]. However, most of these papers were either descriptive or use simulations only. In addition, none of these papers [101-112] has analyzed the router performance in case of MPLS or IP Multicast when DiffServ is adopted, which should raise the level of the thesis research importance.

### **3.8 Conclusions**

In this chapter, a survey of multicasting technologies that are related to thesis work was summarized. This chapter addressed the following subjects: congestion control, reliable IP multicast, MPLS multicast, DiffServ Multicast, DiffServ/MPLS multicast and

finally QoS in heterogeneous networks.

The design and management of such a network is a fundamental key to the success of the QoS provisioning and it includes several open research areas. Many problems need to be solved such as DiffServ multicasting, MPLS multicasting, DiffServ/MPLS multicasting, QoS in heterogeneous networks, LSP dimensioning, set-up/tear-down procedures, routing, adaptation to actual carried traffic, preemption, initial definition of the network topology, etc.

# CHAPTER 4 QoS MULTICAST FOR DIFFSERV OVER MPLS AND IP HOMOGENEOUS NETWORKS

## 4.1 Introduction

Multicasting has been at the center of interest in the Internet area and has already attained major successes. IP Multicast supports group communications by enabling sources to send a single copy of a message to multiple recipients at different locations who explicitly want to receive the information [1]. With the huge increase demand for bandwidth, one of the challenges the Internet is facing today is to boost the packet forwarding performance.

Recent developments in Multiprotocol label Switching (MPLS) open new possibilities to address some of the limitations of IP systems. MPLS is an Internet Engineering Task Force (IETF) standard [10]. It replaces the IP forwarding by a simple label lookup mechanism. MPLS combines the flexibility of layer 3 (L3) routing and layer 2 (L2) switching, which enhances network performance in terms of scalability, computational complexity, latency and control message overhead. Besides this, MPLS offers a vehicle for enhanced network services such as Quality of Services (QoS)/ Class of Service (CoS), Traffic Engineering and Virtual Private Networks (VPNs). IP multicast in MPLS networks is still an open issue [10-25].

On the other hand, the IETF DiffServ working group is looking at a more scalable model and more likely to be easier to implement than IntServ/RSVP model [42]. In the DiffServ architecture, traffic that requires the same Per-Hop-Behavior (PHB) is aggregated into a single queue. The DiffServ architecture focuses on the use of DiffServ (DS) byte, which is the redefined 8-bit Type of Service (TOS) field in the IPv4 header or

the IPv6 Traffic Class octet as a QoS mechanism. Packets are classified into the corresponding queues using their DiffServ Code Points (DSCP). Packets use DSCP bits in order to receive a particular PHB, or forwarding treatment. Marking, classification, traffic conditioning or policing are done at network boundaries (first router for example) and packet treatment and handling is carried on each network node [42-48].

Reliable multicasting is used to provide QoS in group communications for real time multimedia applications such as video conferencing. Two main error control strategies are well known. These are the **FEC (Forward Error Correction)** strategy, which uses error correction alone, and the **ARQ (Automatic Repeat Request)** strategy, which uses error detection, combined with retransmission of repair data [49-60].

In ARQ strategy, when an error is detected at the receiver, a request (NAK) is transmitted to the sender to repeat the incorrect message, and this continues until the message is received correctly. ARQ can be divided into two types: stop-and-wait ARQ and Continuous ARQ, which can be further, divided into two subtypes: go-back-N ARQ and selective-repeat ARQ. In our work, we will use selective repeat ARQ. In addition to that, we will evaluate the performance of the ARQ with both multicast and unicast repairs.

In this chapter, we compare QoS performance of IP and MPLS multicasting in two cases, given their particular constraints [113-114]. Section 4.2 will compare QoS performance of IP and MPLS multicast for a single router case with no reliability consideration and section 4.3 will do a similar thing but for homogeneous multicast networks and where reliability is adopted. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control

packets. We present a new fair share policy and by taking the above constraints into consideration, we evaluate the QoS performance for a typical binary tree in the two cases of IP and MPLS multicasting. We also consider Differentiated Services; i.e. traffics with different priority classes when reliable multicast is used. Analysis tools will be used to evaluate our fair share policy (FSP) for different homogeneous network scenarios.

## **4.2 The Analytical Model Underlying the Fair Share Policy (FSP)**

In this section, a comparison of QoS performance between IP and MPLS multicast for a single router with no reliability adoption is carried out. FSP is not a call admission rather it is a traffic policing mechanism [115]. In FSP, packets are discarded in case of congestion differently at each queue according to source priority and the maximum number in the queue; i.e. the source with higher priority will experience less packet discarding than sources with lower priorities. Moreover, FSP guarantees fairness among flows having the same priority (i.e., required QoS) in buffer space allocated to lower priority traffic is larger; thus leading to less packet discard [113-114]. Our analytical model is shown in Fig. 4-1. In this model, a typical IP or MPLS router and our FSP traffic policing mechanism process three independent sources corresponding to different input traffic classes. Source 1 is assigned the highest priority, then source 2 and finally source 3. For this model, the enforcement is assumed to occur at the router (node) according to Fair Share Policy. The following assumptions are used:

1- The arrival of packets in queue of each priority class is modeled by a discrete time Markovian chain. The time between the states (state transition time) is the service time of a packet which is assumed to be a small and constant  $\Delta T$  (i.e. short discrete intervals). Therefore, it can represent a Poisson arrival such that at most one packet arrives while one packet is served by the line. For example source 1 arrival probability can be defined as:

$$\alpha_1 = \lambda_1 \Delta T \quad \text{where } \lambda_1 \text{ is source 1 arrival rate}$$

2- FSP uses non pre-emptive priority queuing and FIFO for the same priority packets.

3- The arrival probabilities are  $\alpha_1, \alpha_2$  and  $\alpha_3$  for each source respectively. Note that  $\alpha$  represents the probability of a receiving packet while one packet is being served in the channel. In addition to that, assume all packets are of the same length.

4- Service probabilities for different queues are  $\beta_1, \beta_2$  and  $\beta_3$  for each source respectively, which take the priorities into account; i.e. during any packet time server is available only for one class as will follow shortly.

5- Average queue sizes are  $E_1(n), E_2(n)$  and  $E_3(n)$  for each source respectively.

6- Maximum buffer sizes are  $\max_1, \max_2$  and  $\max_3$  for each source respectively.

7- Total system (router) buffer size:  $B = \max_1 + \max_2 + \max_3$  where  $\max_p$  and  $p=1,2,3$  is calculated as:  $\max_p = \frac{Pr_p}{\sum_p Pr_p} * B$  ;  $Pr_p$  is source p priority.

8- All of MPLS or IP routers on the subject Internet are homogeneous in providing resource and traffic conditions, so we take one of them as a representative for IP routers and another one as a representative for MPLS routers.



9- Steady state conditions prevail such that the distribution of the number of packets in the queue will not change with time and hence  $E_1(n)$  for source 1 for example will be taken as a representative figure of the actual number in the queue  $n_1$ .

10- Server is available with probability  $P_c < 1$  due to both errors and losses on the networks; i.e.  $P_c = \mu\Delta T$  where  $\mu$  is the service rate.

11- During a certain packet time (state transition period), one packet may arrive at a certain priority class queue and one packet may be simultaneously served. This is one of the differences from classic M/M/1 systems. Similarly, during a certain packet time there may be no arrival and no service to any of the priority queues [113-114].

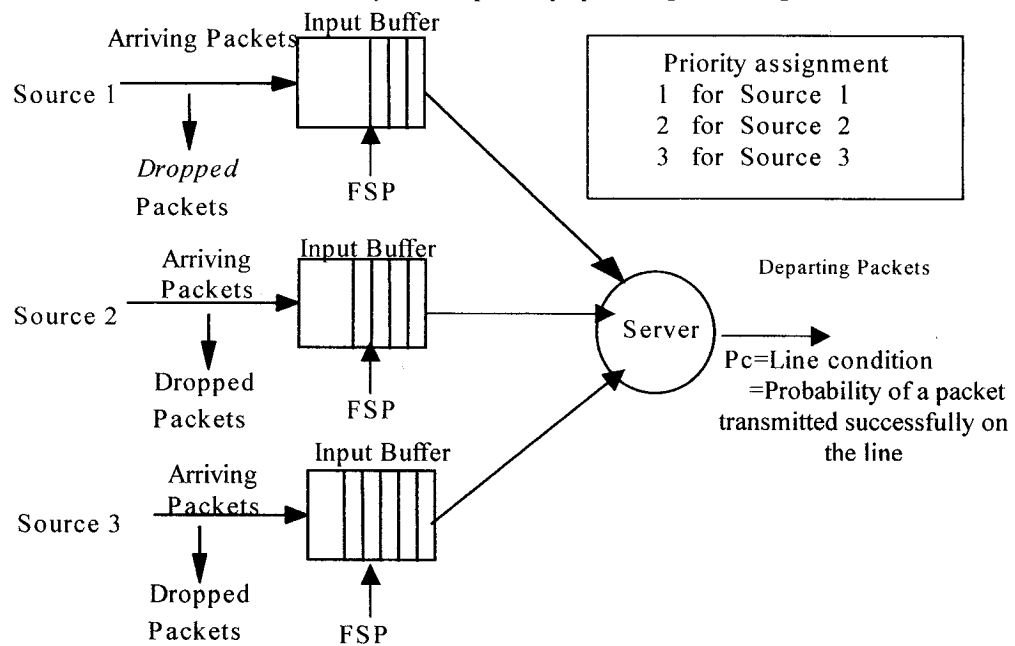


Fig. 4-1 The analytical model

Fig. 4-1 explains the main components of the analytical model for a typical router (IP or MPLS). FSP sets the following rule: low priority users will generally have higher buffer occupancy  $E(n)$  which provides the low priority traffic with more space as needed.

The classifier, which is not shown, aggregates all users' traffics of a certain priority and sends them into the same priority queue.

The coupled discrete Markovian state diagrams for the analytical model in Fig. 4-1 are shown in Fig. 4-2. These diagrams represent a typical router with 3 priority classes. The solution of the number in every class depends on the solutions of the other classes.

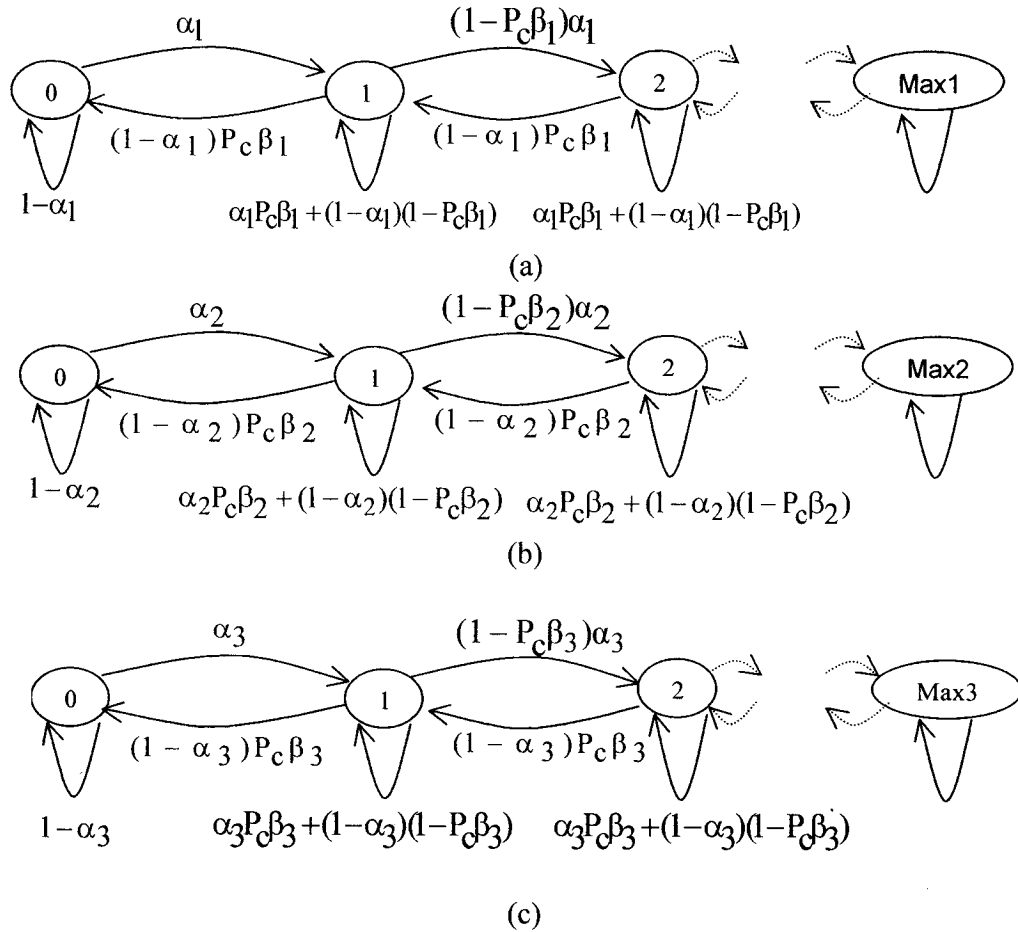


Fig. 4-2 The Coupled Discrete Markovian State Diagrams

To explain some of the state diagram's transitions above, we see in Fig. 4-2 (a) that the transition probability from state 0 to state 1 is the probability of a new arrival from source 1 which is  $\alpha_1$ . Also in the same figure, the transition probability from state 1 to

state 0 is given by  $(1 - \alpha_1)P_c\beta_1$  which is the product of two probabilities, the first one is probability of no new arrivals from source 1  $(1 - \alpha_1)$  and the second one is the probability of serving one packet from source 1  $P_c\beta_1$  (i.e. no arrival from source 1 and one packet is served from source 1). Note that  $\beta_1 = 1$  always since it is the highest priority traffic. In Fig. 4-2 (b), the transition probability from state 1 to state 2 is given by:  $(1 - P_c\beta_2)\alpha_2$  which is the product of two probabilities, the first one is the probability of no packet got served from source 2  $(1 - P_c\beta_2)$  and the second one is the probability of one packet arrival from source 2  $\alpha_2$ . Note that  $\beta_2 = P_0^1$  i.e. packets from source 2 will be served only when the buffer corresponding to source 1 (which has higher priority) is empty.

We see in Fig. 4-2 (c) that the transition probability from state 2 to state 2 (self looping) is given by  $\alpha_3P_c\beta_3 + (1 - \alpha_3)(1 - P_c\beta_3)$  which is the sum of two probabilities, the first one is the probability of one packet arrival from source 3 while one packet is served from source 3  $\alpha_3P_c\beta_3$  and the second one is the probability of no new packet arrival from source 3 and no packet got served from same source  $(1 - \alpha_3)(1 - P_c\beta_3)$ , i.e. this transition probability from state 2 to state 2 is actually the probability of one arrival from source 3 while serving one packet or the probability of no arrivals from source 3 and no packets are being served from the same source. Note that  $\beta_3 = P_0^1P_0^2$  which means packets from source 3 will be served only when the buffers corresponding to source 1 and source 2 (which have higher priority) are all empty.

For IP based networks, the source arrival probability  $\alpha$  is actually a composite one; for instance  $\alpha_1$  can be written as [113-114]:

$$\alpha_1 = \tau\alpha_1^1 + \alpha_1^2, \quad \tau = \frac{\Delta_1 + \Delta_2}{\Delta_1}$$

where  $\Delta_1$  is the processing time at lower layers (for example MAC layer) and  $\Delta_2$  is the processing time at IP layer and  $\tau$  is the IP processing time factor (or processing factor);  $\alpha_1^1$  is the intrinsic arrival probability at the application layer (on top of IP layer),  $\alpha_1^2$  is the extra arrival probability due to IP control overhead which is used to establish the IP multicast tree. The above equation can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad (4-1)$$

where  $\xi_1$  is the IP control overhead factor (or IP factor).

Similarly for MPLS based networks,  $\alpha_1$  can be written as [113-114]:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3$ , where  $\alpha_1^1$  and  $\alpha_1^2$  are the same as in the case of IP networks;  $\alpha_1^3$  is the extra arrival probability due MPLS control overhead which is used to establish the MPLS multicast paths or tree.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2)\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad \xi_2 = \frac{\alpha_1^3}{\alpha_1^1} \quad (4-2)$$

Where  $\xi_2$  is the MPLS control overhead factor (or MPLS factor).

By writing, the balance equations for the state diagram in Fig. 4-2 (a) [116-123], notice that  $\alpha = \alpha_1$ ,  $\beta = \beta_1$  and  $\max = \max_1$ , one finds that

$$P_1 = \frac{\alpha}{\mu} P_0 \quad (4-3)$$

$$P_2 = \frac{1 - \sigma}{\mu} P_1 - \frac{\alpha}{\mu} P_0 = \left[ \frac{(1 - \sigma)\alpha}{\mu^2} - \frac{\alpha}{\mu} \right] P_0 \quad (4-4)$$

$$P_n = \frac{1 - \sigma}{\mu} P_{n-1} - \frac{\lambda}{\mu} P_{n-2} \quad \text{for } n = 3, 4, \dots, \max \quad (4-5)$$

Where  $\lambda = (1 - P_c\beta)\alpha$ ,  $\mu = (1 - \alpha)P_c\beta$  and  $\sigma = \alpha P_c\beta + (1 - \alpha)(1 - P_c\beta)$

Equation (4-5) can be rewritten as:

$$P_{n+2} = \frac{1 - \sigma}{\mu} P_{n+1} - \frac{\lambda}{\mu} P_n \quad \text{for } n = 3, 4, \dots, \max \quad (4-6)$$

Define  $m = \frac{1 - \sigma}{\mu}$  and  $q = \frac{\lambda}{\mu}$

Equation (4-6) is a 2<sup>nd</sup> order homogeneous difference equation [124], which has the general form:

$$P_{n+2} + 2aP_{n+1} + bP_n = 0 \quad \text{for } n=3,4,\dots,\max \quad (4-7)$$

Where  $a = \frac{-m}{2}$  and  $b = q$ , the general solution of equation (4-7) is of the form[124]:

$$P_n = Ar_1^n + Br_2^n \quad \text{for } n=3,4,\dots,\max \quad (4-8)$$

Where  $r_1$  and  $r_2$  are the distinct roots of the equation (4-7) and A and B are constants. The characteristic equation of equation (4-6) is:

$$r^2 - mr + q = 0 \quad (4-9)$$

$$\text{Which has the solution: } r_1 = \frac{m + \sqrt{m^2 - 4q}}{2}, r_2 = \frac{m - \sqrt{m^2 - 4q}}{2}$$

The initial conditions for the set of equations are  $P_1$  and  $P_2$ . Using equation (4-8), we write:

$$P_1 = Ar_1 + Br_2 = kP_0 \quad (4-10)$$

$$P_2 = Ar_1^2 + Br_2^2 = \omega P_0 \quad (4-11)$$

Where  $\omega = \left( \frac{(1-\sigma)\alpha}{\mu^2} - \frac{\alpha}{\mu} \right)$  and  $k = \frac{\alpha}{\mu}$ . Substituting for  $r_1$  and  $r_2$  and solving equations (4-10) and (4-11) together to find A and B, we obtain:

$$B = \left( \frac{\omega - kr_1}{r_2^2 - r_1r_2} \right) P_0 \quad \text{and} \quad A = \left( \frac{kr_1r_2^2 - r_1r_2\omega}{r_1^2r_2^2 - r_1^3r_2} \right) P_0$$

In order to find nth probability  $P_n$ , our solution for equation (4-8) can be written as:

$$P_n = \left( \frac{kr_1r_2^2 - r_1r_2\omega}{r_1^2r_2^2 - r_1^3r_2} \right) P_0 \left( \frac{m + \sqrt{m^2 - 4q}}{2} \right)^n + \left( \frac{\omega - kr_1}{r_2^2 - r_1r_2} \right) P_0 \left( \frac{m - \sqrt{m^2 - 4q}}{2} \right)^n \quad n=3,4,\dots,\max \quad (4-12)$$

Taking into account that  $\sum_{n=0}^{\max} P_n = 1$ ,  $P_0$  can be found using the following equation:

$$P_0 = \frac{1}{1 + \sum_{n=1}^{\max} P_n} = \frac{1}{1 + k + \omega + Ar_1^3 + Br_2^3 + Ar_1^4 + Br_2^4 + \dots + Ar_1^{\max} + Br_2^{\max}} \quad (4-13)$$

$$= \frac{1}{1 + k + \omega + Ar_1 \frac{1 - r_1^{\max}}{1 - r_1} - Ar_1 - Ar_1^2 + Br_2 \frac{1 - r_2^{\max}}{1 - r_2} - Br_2 - Br_2^2}$$

Therefore, the solution of probability of steady state of the number of packets in the buffer is now given by equation (4-12). The expected number of packets in the buffer for a specific source can be found as:

$$E(n) = \sum_{n=0}^{\max} n * P_n = 1 * k * P_0 + 2 * \omega * P_0 + \sum_{n=3}^{\max} n * (Ar_1^n + Br_2^n) \quad (4-14)$$

Notice that the loss probability is equal to the probability of the last state of the state diagram; therefore the loss probability for sources 1, 2 and 3 respectively:

$$P_{L1} = P_{\max 1}, \quad P_{L2} = P_{\max 2} \quad \text{and} \quad P_{L3} = P_{\max 3} \quad (4-15)$$

The same solution above applies to the state diagrams in Figs. 4-2 (b) and 4-2 (c) as well except that in Fig. 4-2 (b)  $\alpha = \alpha_2$ ,  $\beta = \beta_2$  and  $\max = \max_2$  and that in Fig. 4-2 (c)  $\alpha = \alpha_3$ ,  $\beta = \beta_3$  and  $\max = \max_3$

### 4.2.1 Analysis Results

Fig. 4-3 shows the expected number of packets in a typical router buffer versus IP factor  $\xi_1$  for all sources for both IP and MPLS. The figure shows that IP and MPLS have very similar expected number of packets especially for low priority traffic and when the intrinsic arrival rates are relatively high. Note that the value of processing factor ( $\tau$ ) is

relatively small meaning that the difference in packet processing between IP and MPLS is small.

Fig. 4-4 shows the packet loss probability for all sources for both IP and MPLS versus IP factor for relatively smaller intrinsic arrival rates and small processing factor ( $\tau$ ). It shows that IP and MPLS have almost the same loss probability, except a small difference for source 3; and as IP factor increases the difference becomes even smaller.

However, Figs. 4-5 and 4-6 show that when the processing factor ( $\tau$ ) increases MPLS will have superiority over IP in terms of the expected number of packets in the router buffer and packet loss probability. As shown in Fig. 4-5, the expected number of packets in the router buffer in case of MPLS is less than IP for all sources and this difference is clear for low priority sources 2 and 3. Fig. 4-6 shows that the packet loss probability in the case of MPLS is less than IP for all sources. This means when the difference in packet processing ( $\tau$ ) between MPLS and IP increases, MPLS will be better.

In the previous Figs. 4-3, 4-4, 4-5 and 4-6 MPLS factor  $\xi_2$  was constant and relatively small; explaining why MPLS performance was better or very similar to IP performance. However, in the following figures we will study the effects of MPLS factor on MPLS performance. Figs. 4-7 and 4-8 show that IP will be superior over MPLS when MPLS factor increases. As shown in Fig. 4-7, the expected number of packets in the typical router buffer in the case of IP (which is constant) is less than MPLS. Similarly, Fig. 4-8 shows that the packet loss probability in the case of IP (which is constant) is less than MPLS. This means when the extra arrival rate due MPLS control overhead used to establish MPLS multicast paths or tree increases, IP will be perform better.

Figs. 4-9 to 4-16 are drawn with high arrival rates and high  $P_c$  for all sources while Figs. 4-17 to 4-20 are drawn with low arrival rates and high  $P_c$  for all sources

Figs. 4-9 and 4-10 show the IP expected number of packets in the router buffer for sources 1 and 2 respectively versus IP factor and processing factor, while Figs. 4-11 and 4-12 show the MPLS expected number of packets in the buffer for sources 1 and 2 respectively versus IP factor and MPLS factor. As shown in Figs. 4-9, 4-10, 4-11 and 4-12 for sources 1 and 2, IP will perform better than MPLS in terms of the expected number of packets in the router buffer if the processing factor is small and if the MPLS factor is large. However, MPLS will perform better than IP if the processing factor is large and if MPLS factor is small.

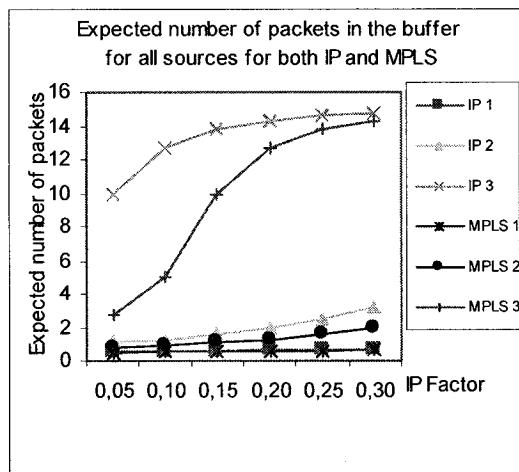
Figs. 4-13 and 4-14 show the IP packet loss probability for sources 1 and 3 respectively versus IP factor and processing factor, while Figs. 4-15 and 4-16 show the MPLS packet loss probability for sources 1 and 3 respectively versus IP factor and MPLS factor. As shown in Figs. 4-13 and 4-15 for source 1 IP will perform better than MPLS in terms of packet loss probability if the processing factor is small and if MPLS factor is large. However, MPLS will perform better than IP if the processing factor is large and if MPLS factor is small. One may notice that in Figs. 4-14 and 4-16 for source 3 (lowest priority traffic) IP and MPLS perform similarly.

Fig. 4-17 shows the IP expected number of packets in the router buffer for source 1 versus IP factor and processing factor, while Fig. 4-18 shows the MPLS expected number of packets in the buffer for source 1 versus IP factor and MPLS factor. As shown in Figs. 4-17 and 4-18 for source 1, IP will perform better than MPLS in terms of expected number of packets in the router buffer if the processing factor is small and if MPLS factor



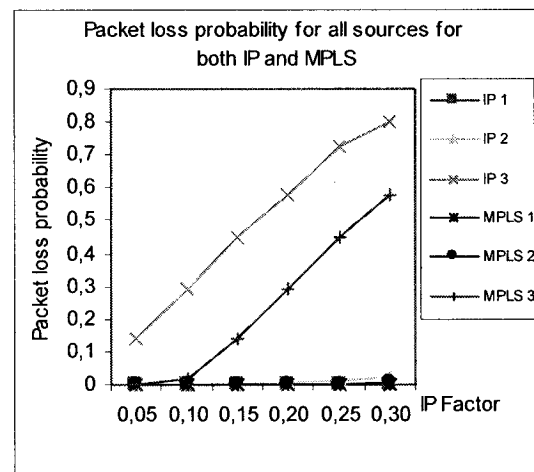
is large. However, MPLS will perform better than IP if the processing factor is large and if MPLS factor is small.

Fig. 4-19 shows the IP packet loss probability for source 3 versus IP factor and processing factor, while Fig. 4-20 shows the MPLS packet loss probability for source 3 versus IP factor and MPLS factor. As shown in the figures, the packet loss probability is very small for all IP and MPLS sources because of the low arrival rates. In addition, these figures show that IP will perform better than MPLS in terms of packet loss probability if the processing factor is small and if MPLS factor is large. However, MPLS will perform better than IP if the processing factor is large and if MPLS factor is small.



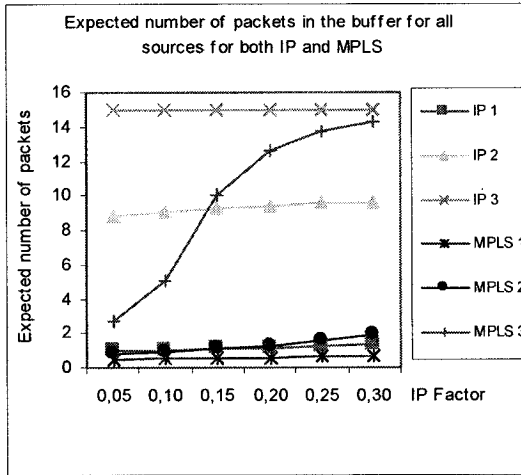
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $P_c = 0.8, B = 30, \xi_2 = 0.1, \tau = 1.2$

Fig. 4-3 Expected number of packets in the buffer for all sources for both IP and MPLS (small  $\tau$ )



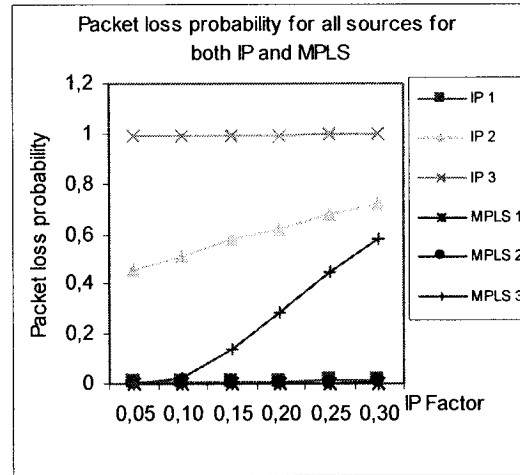
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $P_c = 0.8, B = 30, \xi_2 = 0.1, \tau = 1.2$

Fig. 4-4 Packet loss probability for all sources for both IP and MPLS (small  $\tau$ )



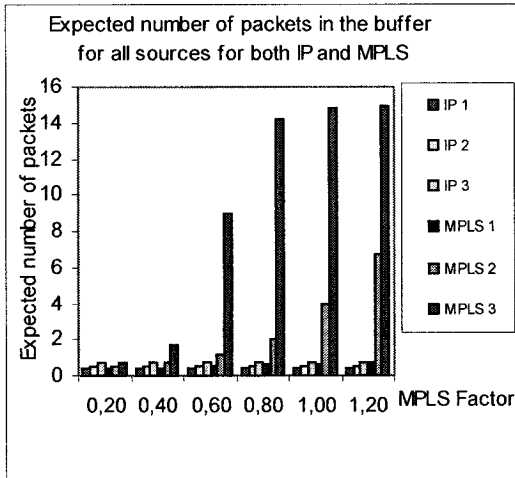
$$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1, \\ P_c = 0.8, B = 30, \xi_2 = 0.1, \tau = 1.8$$

Fig. 4-5 Expected number of packets in the buffer for all sources for both IP and MPLS (large  $\tau$ )



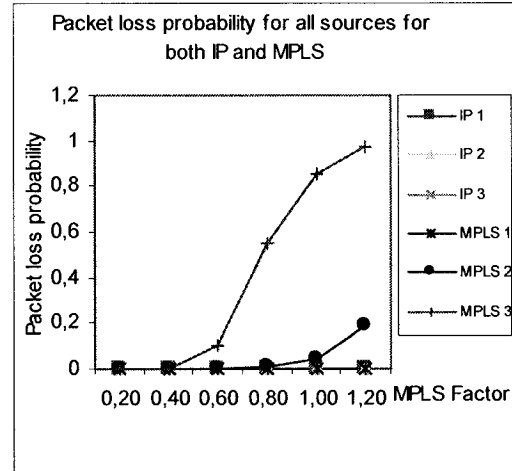
$$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1, \\ P_c = 0.8, B = 30, \xi_2 = 0.1, \tau = 1.8$$

Fig. 4-6 Packet loss probability for all sources for both IP and MPLS (large  $\tau$ )



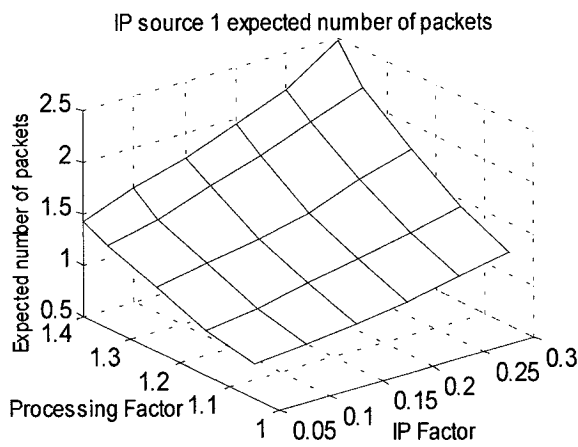
$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30, \xi_1 = 0.2, \tau = 1.2$$

Fig. 4-7 Expected number of packets in the buffer for all sources for both IP and MPLS (effect of MPLS factor)



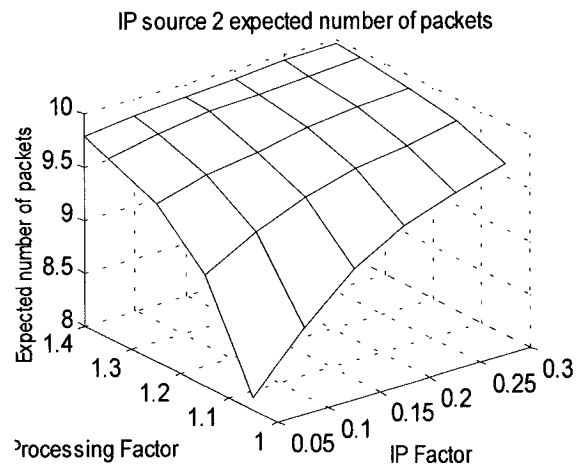
$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30, \xi_1 = 0.2, \tau = 1.2$$

Fig. 4-8 Packet loss probability for all sources for both IP and MPLS (effect of MPLS factor)



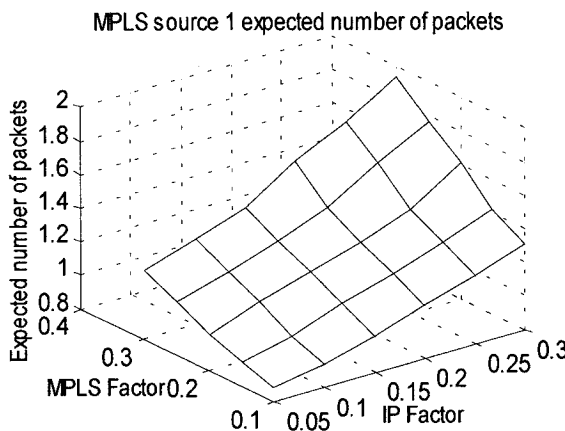
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-9 IP Expected number of packets in the buffer for source 1 versus IP factor and processing factor (high arrival rates)



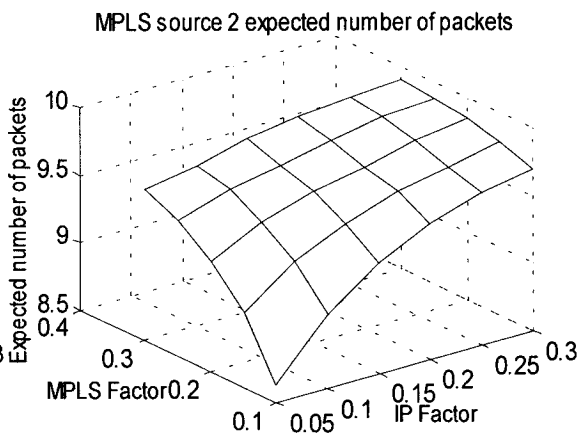
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-10 IP Expected number of packets in the buffer for source 2 versus IP factor and processing factor (high arrival rates)



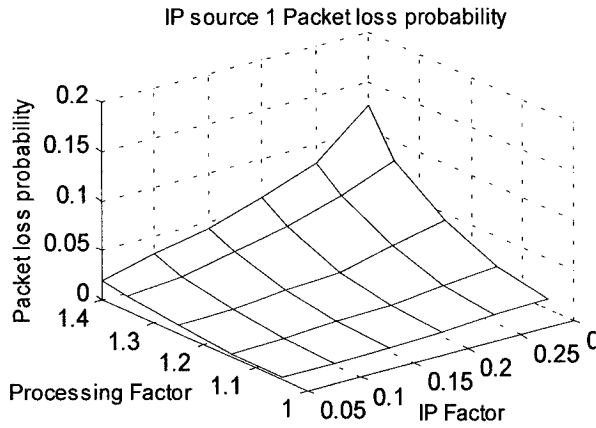
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-11 MPLS Expected number of packets in the buffer for source 1 versus IP factor and MPLS factor (high arrival rates)



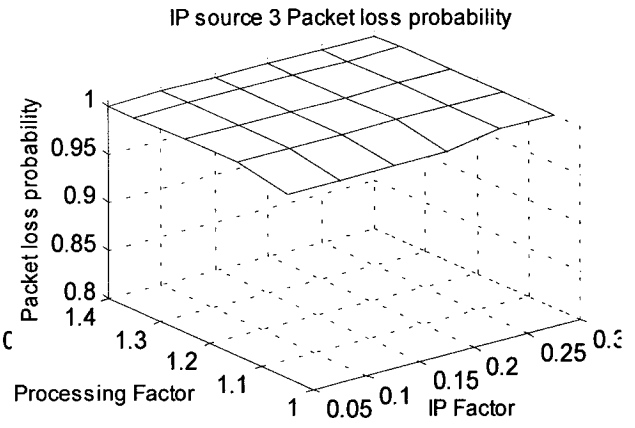
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-12 MPLS Expected number of packets in the buffer for source 2 versus IP factor and MPLS factor (high arrival rates)



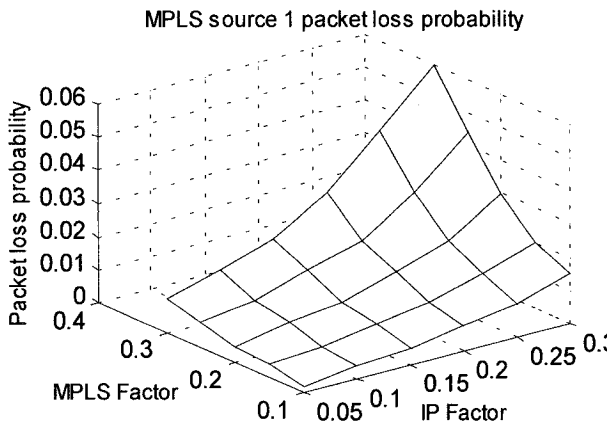
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-13 IP Packet loss probability for source 1 versus IP factor and processing factor (high arrival rates)



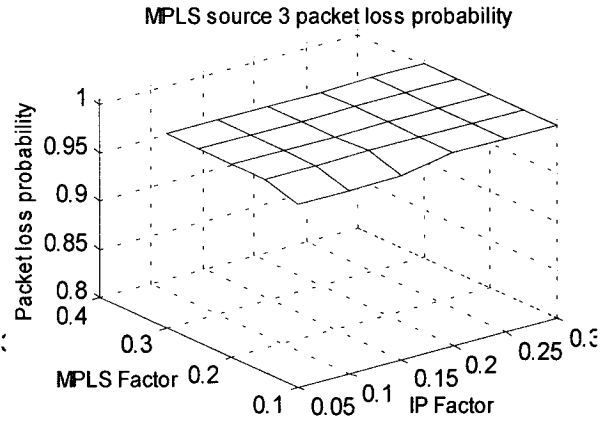
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-14 IP Packet loss probability for source 3 versus IP factor and processing factor (high arrival rates)



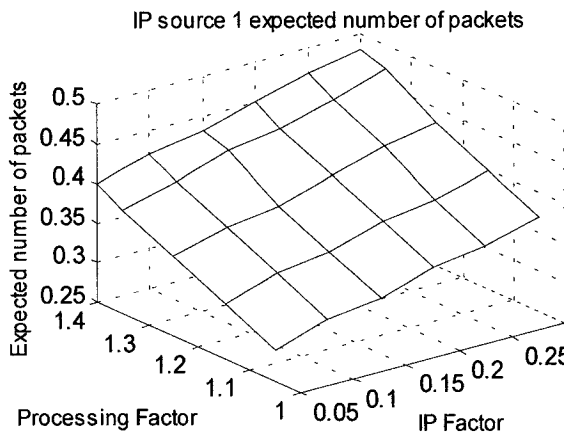
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig. 4-15 MPLS Packet loss probability for source 1 versus IP factor and MPLS factor (high arrival rates)



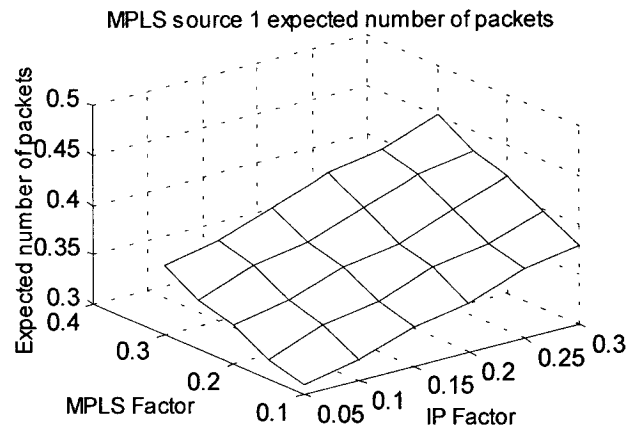
$$\alpha_1^1 = 0.45, \alpha_1^2 = 0.35, \alpha_1^3 = 0.3, \beta_1 = 1, P_c = 0.8, B = 30$$

Fig.4-16 MPLS Packet loss probability for source 3 versus IP factor and MPLS factor (high arrival rates)



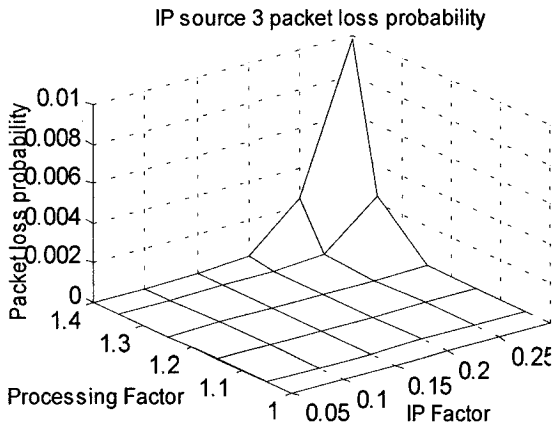
$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30$$

Fig. 4-17 IP Expected number of packets in the buffer for source 1 versus IP factor and processing factor (low arrival rates)



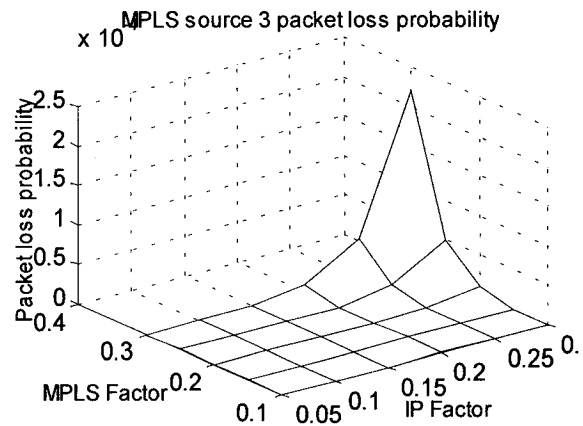
$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30$$

Fig. 4-18 MPLS Expected number of packets in the buffer for source 1 versus IP factor and MPLS factor (low arrival rates)



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30$$

Fig. 4-19 IP Packet loss probability for source 3 versus IP factor and processing factor (low arrival rates)



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, \\ P_c = 0.8, B = 30$$

Fig. 4-20 MPLS Packet loss probability for source 3 versus IP factor and MPLS factor (low arrival rates)

## 4.3 Homogeneous Reliable Multicast Tree

In this section, a comparison of QoS performance between IP and MPLS multicast for a homogeneous reliable multicast tree is carried out.

### 4.3.1 The Analytical Model Underlying the Fair Share Policy (FSP)

The same analytical model shown in Fig. 4-1 and the same discrete coupled state diagrams shown in Fig. 4-2 will be used through out the analysis part of this thesis, i.e.; in chapters 4, 5, and 6.

One more assumption is added for reliable multicast that a complete homogeneous binary multicast tree would be used, where each parent router has two children routers until we reach leafs [114]. Fig. 4-21 shows an example of a complete binary multicast tree with the root, which is the nearest router or node to the sender or the rendezvous point, and the leafs, which are the routers with receivers underneath them. As shown in Fig. 4-21 the depth of this tree is 4 and the total number of routers is 15.

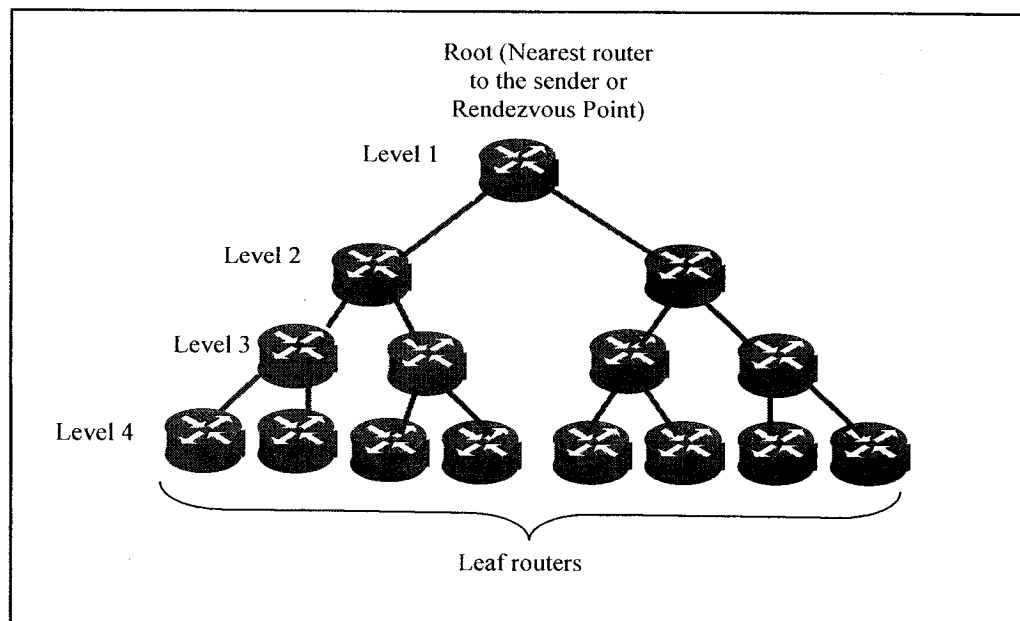


Fig. 4-21 A complete homogeneous binary multicast tree

### 4.3.2 Reliable Multicast Cases Under Study

In this section, the following four cases will be studied :

#### Case 1: Without FEC or ARQ

In the first case, we make a performance comparison between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ is adopted and when reliability is not considered, i.e., no FEC or ARQ would be used.  $P_{c_p}$ , which is the probability of successful delivery to next router for certain priority traffic  $p$ , would be given as:

$$P_{c_p} = (1 - P_{o_p} - P_{e_p})^L, \quad p = 1, 2 \text{ or } 3 \quad (4-16)$$

Where  $P_{o_p}$ , is the byte overflow (loss) for a certain priority traffic and is given by:

$$P_{o_p} = \frac{\text{PacketLoss (Overflow) Probability}}{L}, \quad p = 1, 2 \text{ or } 3 \quad (4-17)$$

$P_{e_p}$  is the byte error probability for certain priority traffic and  $L$  is the number of bytes per packet. In the previous equations, two assumptions are made:

- 1- Packet loss of source packet is caused by consecutive byte losses at the intermediate routers.
- 2- Interleaving is used in order to break byte burst losses and efficiently turn them independent random byte losses at the source and destination [125].

Probability of no packet loss for certain priority traffic is given by:

$$P_{\text{no packet loss}_p} = (1 - P_{o_p})^L \cong 1 - LP_{o_p} \quad \text{for small values of } P_{o_p}, \quad p = 1, 2, 3$$

Therefore, probability of packet loss for certain priority traffic can be expressed as:

$$P_{\text{packet loss}_p} = 1 - P_{\text{no packet loss}_p} = LP_{o_p} \quad \text{for small values of } P_{o_p}, \quad p = 1, 2, 3$$

$$\therefore P_{o_p} = \frac{P_{\text{PacketLoss}_p}}{L}, \quad p = 1, 2, 3$$

The total delay a certain packet with priority  $p$  would experience from sender until it reaches the receiver is given by:

$$D_{p \text{ Total}} = D \overline{D_p}, \quad p = 1,2,3 \quad (4-18)$$

Where  $D$  is the number of routers in the longest path (Depth) to the receivers (leafs of the tree), and  $\overline{D_p}$  is the average packet queuing delay (or expected number of packets) per router in terms of packets for certain priority traffic  $p$ . The expected number of packets for traffic with priority  $p$  can be found using:

$$\overline{E_p}(n) = \sum_{i=1}^{\max} i P_i \quad p=1,2,3 \quad (4-19)$$

Equation 4-19 actually can be evaluated using equation 4-14. The second moment of delay in units of (Packets)<sup>2</sup> can be found using:  $\overline{E_p^2}(n) = \sum_{i=1}^{\max} i^2 P_i, \quad p = 1,2,3$

Delay jitter (standard deviation) per router for a certain priority  $p$  can be expressed as:

$$\sigma_{xp} = \sqrt{\overline{E_p^2}(n) - (\overline{E_p}(n))^2}, \quad p = 1,2,3$$

By assuming that total delays for all routers are statistically independent, the total delay jitter for certain priority traffic (total standard deviation) can be found using:

$$\sigma_{p \text{ Total}} = D \sigma_{xp}, \quad p = 1,2,3 \quad (4-20)$$

Probability of multicast success (all  $N$  routers receive the multicast packet) and multicast residual packet loss probability for certain priority traffic can be found using the following equations:

$$P_{s_p} \equiv \text{Probability of success} = P_{c_p}^N \quad \text{and} \quad P_{loss_p} = 1 - P_{s_p}, \quad p = 1,2,3 \quad (4-21)$$

Where  $N$  is total number of routers in the multicast network.



### Case 2 Using FEC only

In the second case, we make a performance comparison between IP multicast in MPLS networks using FSP and plain IP multicasting using the same policy when DiffServ is adopted and when reliable multicast using FEC only is assumed. There are  $\binom{L}{i}$  ways to have errors in  $i$  bytes out of  $L$  bytes in the multicast packet received at a certain router. However, once we have  $i$  bytes in errors, the number of ways of selecting the location of lost bytes would be  $\binom{L-i}{j}$ .  $P_{c_p}$  which is the probability of successful delivery to next router for certain priority traffic  $p$ , is then given by:

$$P_{c_p} = \sum_{i=0}^L \sum_{j=0}^{L-i} \binom{L}{i} P_{e_p}^i \binom{L-i}{j} P_{o_p}^j (1 - P_{o_p} - P_{e_p})^{L-i-j}, p = 1,2,3 \quad (4-22)$$

Provided that  $2i+j \leq e = n-k$ ; where  $r$  is the FEC code rate;  $r=k/n$ ,  $k$  is the number of original data symbols,  $n$  is the total number of symbols after applying FEC encoding.  $P_{o_p}$  is the byte overflow for a certain priority traffic and  $P_{e_p}$  is the byte error probability for a certain priority traffic. Notice that due to FEC use, the intrinsic arrival probability  $\alpha_p^1$  increases to:

$$\alpha_p^{1'} = \frac{1}{r_p} \alpha_p^1, p = 1,2, 3, \dots \text{ i.e., \# of priority classes} \quad (4-23)$$

The total delay and total delay jitter can be found using equations 4-18 and 4-20 from Case 1. Probability of multicast success (all  $N$  routers receive the multicast packet) and residual multicast loss probability for certain priority traffic  $p$  are given as:

$$P_{s_p} \equiv \text{Probability of success} = P_{c_p}^N \quad \text{and} \quad P_{\text{loss}_p} = 1 - P_{s_p} \quad (4-24)$$

### Case 3 Using ARQ only

In the third case, we make a performance comparison between IP multicast in MPLS networks using FSP and plain IP multicasting when DiffServ is adopted and reliable

multicast using ARQ only assumed. In our work we use selective-repeat ARQ. In case of ARQ only,  $Pc_p$  would be similar to case 1 and is given as:

$$Pc_p = (1 - Po_p - Pe_p)^L, \quad p = 1, 2 \text{ or } 3, \text{ which is the priority of a certain traffic} \quad (4-25)$$

We have two subcases: ARQ only that uses multicast repairs and ARQ only that uses unicast repairs.

#### a) ARQ Multicast repairs

In this case upon the receipt of a NAK from one or more receivers, the sender multicast again the repair packet to all receivers. Due to the use of ARQ multicast repairs, the intrinsic arrival probability  $\alpha_p^1$  for certain priority traffic  $p$  would increase according to:

$$\alpha_p^1 = \alpha_p^1(1 + F_p), \quad p = 1, 2, 3 \quad (4-26)$$

$F_p$  is the number of failures for certain priority traffic  $p$ . This increase in the intrinsic arrival probability is due to that every router in the whole network receives a copy of each repair packet. The Probability of success for worst case scenario for certain priority traffic  $p$  is given as:

$$Ps_p \equiv \text{Probability of success} = Pc_p^N \quad (\text{worst case scenario}) \quad (4-27)$$

Equation 4-27 represents an upper bound for worst case scenario of probability of success when ARQ multicast repairs method is used. However, using ARQ multicast repairs have a better chance of success with each trial since the number of receivers who did not receive the packet correctly decreases with each trial. Therefore, the average probability of success for a certain priority  $p$  packet in a typical transmission multicast trial from sender can be calculated as:

$$Ps_{p,avg} = \frac{Pc_p^N + Pc_p^{(N/2)+1} + Pc_p^{(N/4)+2} + Pc_p^{(N/8)+3} + \dots}{D} \quad (4-28)$$

Where  $D$  is the network depth. If the packet does not suffer loss or error on any of the  $N$  routers of the multicast tree, with probability  $P_c^N$  no further repair is needed, this explains the first term of equation 4-28. However, if there has been an error or loss which is located at level  $l$  (see Fig. 4-21), then the repair packet would be sent from sender to the router at level  $l$ , and then the repair packet will flow to  $N/2$  routers under level  $l$ . All such  $(N/2)^{l-1} + 1$  transmissions of repair packet have to be correct, otherwise further repair is needed and so on. The probability of these  $(N/2)^{l-1} + 1$  correct transmissions of subject repair packet is given by  $P_c^{(N/2)^{l-1} + 1}$  and so on for the remaining terms in equation 4-28. We divide by  $D$  (network depth) because we assume that errors are equally likely to occur on different levels of the tree giving rise to the addition of different terms (equation 4-28) and the division by the depth  $D$  where  $D = \log_2(N + 1)$ .

Note that the average  $P_{s_p}$  for ARQ case for certain priority traffic  $p$  deals only with the number of routers that should have correct transmissions (no loss and no errors) during the repair trial of one previous loss depending on the location of this previous loss. For example in level 2 of the multicast tree, the repair packet should be transmitted correctly to the router in question from the sender, this means two correct transmissions of the repair packet (2 links from sender to router that needs repair). Furthermore, the repair packet must be forwarded correctly to all routers of the sub-tree under the router in question, which means  $N/4$  correct transmissions of the repair packet. Needless to say during first transmission or subsequent repair trials of a certain packet, one or more errors or packet losses may take place in different levels or places. This will reflect itself in having a higher number of repair trials demanded by the ARQ process, but each repair trial will face the average  $P_{s_p}$  above. On the other hand one may take  $P_{s_p}$  to be equal to

a worst case scenario; which will lead to unnecessary and unrealistic degradation of performance since the number of nodes needing repairs decrease with each trial (we do not transmit to nodes who have already got the subject packet in early ARQ trials, i.e. corresponding to the latter worst case was shown before in equations 4-21 and 4-24). The total number of ARQ trials  $T_p$  for specific priority traffic  $p$  can be expressed as:

$$T_p \equiv \text{Total number of trials} = Ps_p + 2Ps_p(1 - Ps_p) + 3Ps_p(1 - Ps_p)^2 + \dots \quad (4-29)$$

Therefore, the number of failures (or retransmissions only) for certain priority traffic  $p$  can be given as:

$$F_p = T_p - 1$$

Where  $Ps_p$  is the average probability of packet success for priority  $p$  traffic corresponding to one ARQ trial. Since equation (4-29) is a geometric series, therefore  $F_p$  can be written in a closed form as:

$$\left\{ \begin{array}{l} F_p = \frac{1}{Ps_p} - 1 \text{ for infinite number of ARQ trials} \\ \text{and } F_p = \frac{1 - (z+1)(1 - Ps_p)^z + (z)(1 - Ps_p)^{z+1}}{Ps_p} - 1 \text{ for } z \text{ multicast trials of a certain packet} \end{array} \right. \quad (4-30)$$

Defining  $Ps_p'$  as the final probability of success for priority  $p$  traffic:

$$\begin{aligned} Ps_p' &= Ps_p + (1 - Ps_p)Ps_p + (1 - Ps_p)^2 Ps_p + \dots + Ps_p(1 - Ps_p)^{z-1} \\ &= 1 - (1 - Ps_p)^z \quad \text{for } (z) \text{ trials of a typical packet to the multicast tree.} \end{aligned} \quad (4-31)$$

Where we note that for one trial  $Ps_p' = Ps_p$  and for infinite retransmission trials  $Ps_p' = 1$  as it should be. Therefore, the residual loss (after all ARQ trials) is given by:

$$P_{loss_p} = 1 - Ps_p' \quad (4-32)$$

The Total delay for specific priority traffic is given by:

$$D_{pTotal} = (1 + F_p) \overline{D D_p} \quad (4-33)$$

Where  $F_p$  is the number of failures for priority  $p$  traffic,  $D$  is network depth and  $\overline{D_p}$  is the average packet delay per router for packet with priority  $p$ . The total delay jitter can be found using equation (4-20) from Case 1.

### b) ARQ Unicast repairs

With unicast repairs, if the sender receives a NAK from one or more receivers, it resends the repair packet to only the receivers who did not receive the packet correctly in a unicast manner. The Multicast repairs method is simpler than the unicast repairs method and requires less overhead; however the multicast repairs methods consumes much more bandwidth. In our work we will evaluate the performance of the ARQ with both multicast and unicast repairs. Due to use of ARQ unicast repairs, the intrinsic arrival probability for certain priority traffic  $\alpha_p^1$  would change according to:

$$\alpha_p^{1'} = \alpha_p^1 \left(1 + F_p \frac{D}{N}\right) (1 + \Delta), \quad p = 1,2,3 \quad (4-34)$$

Where  $\Delta$  is the extra arrival rate due to processing of unicast repairs. This increase in the intrinsic arrival probability is due to the fact that only those routers on the path (of maximum  $D$  hops) to the router that requires repair would need the repair packet. These routers only receive a copy of repair packet so the retransmission factor would be  $(D/N)$  as compared to the multicast case where we have  $(N/N)$ .

Fig. 4-22 shows the conditional probabilities tree for unicast repair. As shown in the figure, for one trial that succeeded in the first trial, the probability of success would be  $Pc_p^N$  and for two trials that failed in the first trial and succeeded in the second one, the probability of success would be equal to  $(1 - Pc_p^N)Pc_p^D$  and so on for the rest of the tree.

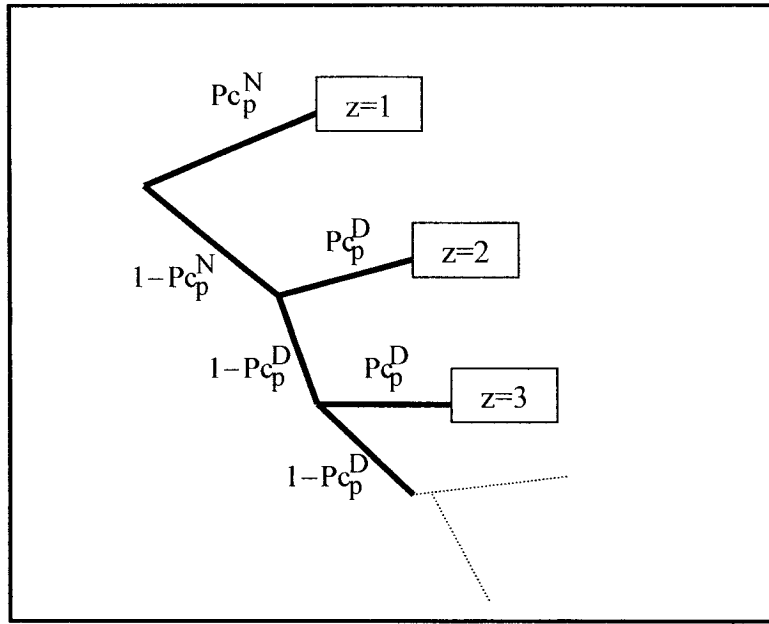


Fig. 4-22 Conditional probabilities tree for unicast repair

The total number of trials  $T_p$  for specific priority traffic  $p$  can be found in a closed form for infinite number of trials as:

$$\begin{aligned}
 T_p &= 1 * Pc_p^N + 2 * (1 - Pc_p^N) Pc_p^D + 3 * (1 - Pc_p^N) (1 - Pc_p^D) Pc_p^D \\
 &\quad + 4 * (1 - Pc_p^N) (1 - Pc_p^D)^2 Pc_p^D + \dots \\
 &= Pc_p^N + (1 - Pc_p^N) \left\{ \sum_{i=1}^{\infty} (i+1) (1 - Pc_p^D)^{i-1} Pc_p^D \right\}
 \end{aligned}$$

The summation in the previous equation is a summation of two geometric series.

Therefore:

$$T_p = Pc_p^N + (1 - Pc_p^N) \left[ 1 + \frac{1}{Pc_p^D} \right] \quad \text{for infinite trials} \quad (4-35)$$

The number of failures  $F_p$  (or retransmissions only) for specific traffic with priority  $p$  is

given as:

$$F_p = T_p - 1 = Pc_p^N + (1 - Pc_p^N) \left[ 1 + \frac{1}{Pc_p^D} \right] - 1 = \frac{1 - Pc_p^N}{Pc_p^D} \quad \text{for infinite trials} \quad (4-36)$$

The total number of trials  $T_p$  for specific priority traffic  $p$  can be found in a closed form for  $(z)$  trials as:

$$T_p = 1 * Pc_p^N + (1 - Pc_p^N) \sum_{i=1}^{z-1} (i+1) (1 - Pc_p^D)^{i-1} Pc_p^D$$

Taking  $Pc_p^D$  as a common factor and dividing by  $(1 - Pc_p^D)^2$ , the previous equation becomes:

$$T_p = Pc_p^N + (1 - Pc_p^N) \frac{Pc_p^D}{(1 - Pc_p^D)^2} \sum_{i=1}^{z-1} (i+1) (1 - Pc_p^D)^{i+1}$$

Substitute  $(i+1)$  by  $j$  will result in the following equation:

$$T_p = Pc_p^N + \frac{(1 - Pc_p^N)Pc_p^D}{(1 - Pc_p^D)^2} \left[ \sum_{j=2}^z j (1 - Pc_p^D)^j \right]$$

By adding and subtracting the first series term; and by separating the last term from series, the previous equation becomes:

$$T_p = Pc_p^N + \frac{(1 - Pc_p^N)Pc_p^D}{(1 - Pc_p^D)^2} \left\{ \left[ \sum_{j=1}^{z-1} j (1 - Pc_p^D)^j \right] + z (1 - Pc_p^D)^z - (1 - Pc_p^D) \right\}$$

By using the arithmetic-geometric series summation, we get:

$$T_p = Pc_p^N + \frac{(1 - Pc_p^N)Pc_p^D}{(1 - Pc_p^D)^2} \left\{ \left[ \frac{(1 - Pc_p^D) \{ 1 - z(1 - Pc_p^D)^{z-1} + (z-1)(1 - Pc_p^D)^z \}}{[Pc_p^D]^2} \right] + z(1 - Pc_p^D)^z - (1 - Pc_p^D) \right\} \quad (4-37)$$

Therefore, the number of failures (or retransmissions only) for certain priority traffic  $p$  for  $(z)$  trials can be given as:

$$F_p = T_p - 1 \quad \text{for } (z) \text{ trials} \quad (4-38)$$

Defining  $Ps_p'$  as the final probability of success for priority  $p$  traffic for infinite number of trials:

$$\begin{aligned} Ps_p' &= Pc_p^N + (1 - Pc_p^N)Pc_p^D + (1 - Pc_p^N)(1 - Pc_p^D)Pc_p^D + (1 - Pc_p^N)(1 - Pc_p^D)^2 Pc_p^D + \dots \\ &= Pc_p^N + (1 - Pc_p^N)Pc_p^D \left\{ 1 + (1 - Pc_p^D) + (1 - Pc_p^D)^2 + \dots \right\} \\ &= Pc_p^N + (1 - Pc_p^N) = 1 \end{aligned} \quad (4-39)$$

Where we note that for one trial of a typical packet to the multicast tree  $Ps_p' = Pc_p^N$  and for infinite retransmission trials  $Ps_p' = 1$  as it should be. For finite number of trials (z),  $Ps_p'$  can be found using:

$$\begin{aligned}
Ps_p' &= Pc_p^N + (1 - Pc_p^N)Pc_p^D + (1 - Pc_p^N)(1 - Pc_p^D)Pc_p^D + \dots + (1 - Pc_p^N)(1 - Pc_p^D)^{z-2} Pc_p^D + \dots \\
&= Pc_p^N + (1 - Pc_p^N)Pc_p^D \frac{1 - (1 - Pc_p^D)^{z-1}}{1 - (1 - Pc_p^D)} \\
&= Pc_p^N + (1 - Pc_p^N) \left[ 1 - (1 - Pc_p^D)^{z-1} \right] \tag{4-40}
\end{aligned}$$

Therefore, the residual loss (after all ARQ trials) is given by:

$$Ploss_p = 1 - Ps_p' \tag{4-41}$$

The total delay for specific priority traffic p is given by:

$$D_{pTotal} = (1 + F_p) \overline{D D_p} \tag{4-42}$$

Where  $F_p$  is the number of failures for priority traffic p,  $D$  is network depth and  $\overline{D D_p}$  is the average packet delay per router for packet with priority p. The total delay jitter can be found using equation (4-20) from Case 1.

#### Case 4 Hybrid FEC/ARQ

A hybrid FEC/ARQ strategy should be used where a combination of FEC for the most frequent error patterns, together with error detection and retransmission for the less likely error patterns is more efficient than ARQ alone. In this case when FEC fails to correct errors at the receiver the receiver sends a NAK to the sender to retransmit the data in error. This hybrid FEC/ARQ strategy clearly carries the potential for improving throughput in two-way systems subject to a high channel error rate. In this case,  $Pc_p$  would be similar to the case of FEC only, which is:



$$P_{c_p} = \sum_{i=0}^L \sum_{j=0}^{L-i} \binom{L}{i} P_{e_p}^i \binom{L-i}{j} P_{o_p}^j (1 - P_{o_p} - P_{e_p})^{L-i-j}, p = 1,2,3 \quad (4-43)$$

Provided that  $2i + j \leq e = n - k$ ;  $r = \frac{k}{n}$ ;  $r$  is the FEC coding rate. The intrinsic arrival probabilities for certain traffic with priority  $p$  ( $p=1,2,3$ ) could be given as:

$$\left\{ \begin{array}{l} \alpha_p^{1'} = \frac{\alpha_p^1}{r_p} (1 + F_p) \quad \text{(For multicast case)} \\ \alpha_p^{1'} = \frac{\alpha_p^1}{r_p} (1 + F_p \frac{D}{N}) (1 + \Delta) \quad \text{(For unicast case)} \end{array} \right. \quad (4-44)$$

Also we have 2 subcases, FEC/ARQ that uses multicast repairs and FEC/ARQ that uses unicast repairs. The analysis of hybrid FEC/ARQ would be very similar to ARQ only case except a better value of  $P_c$  which should be found as specified by equation (4-43). In addition, there would be an increase in the intrinsic arrival probabilities as in equation (4-44). Notice that the total delay jitter can be found using equation (4-20) from Case 1.

Six different programs were developed for the purpose of calculations and to solve the involved and non-linear set of equations in order to find the performance measures (delay, jitter and probability of packet loss probability). These programs are kept running until the set of equations converge. We will explain how the program calculate the performance measures in case of Hybrid unicast repairs since it is the most general case aided with flowchart of Fig. 4-23.

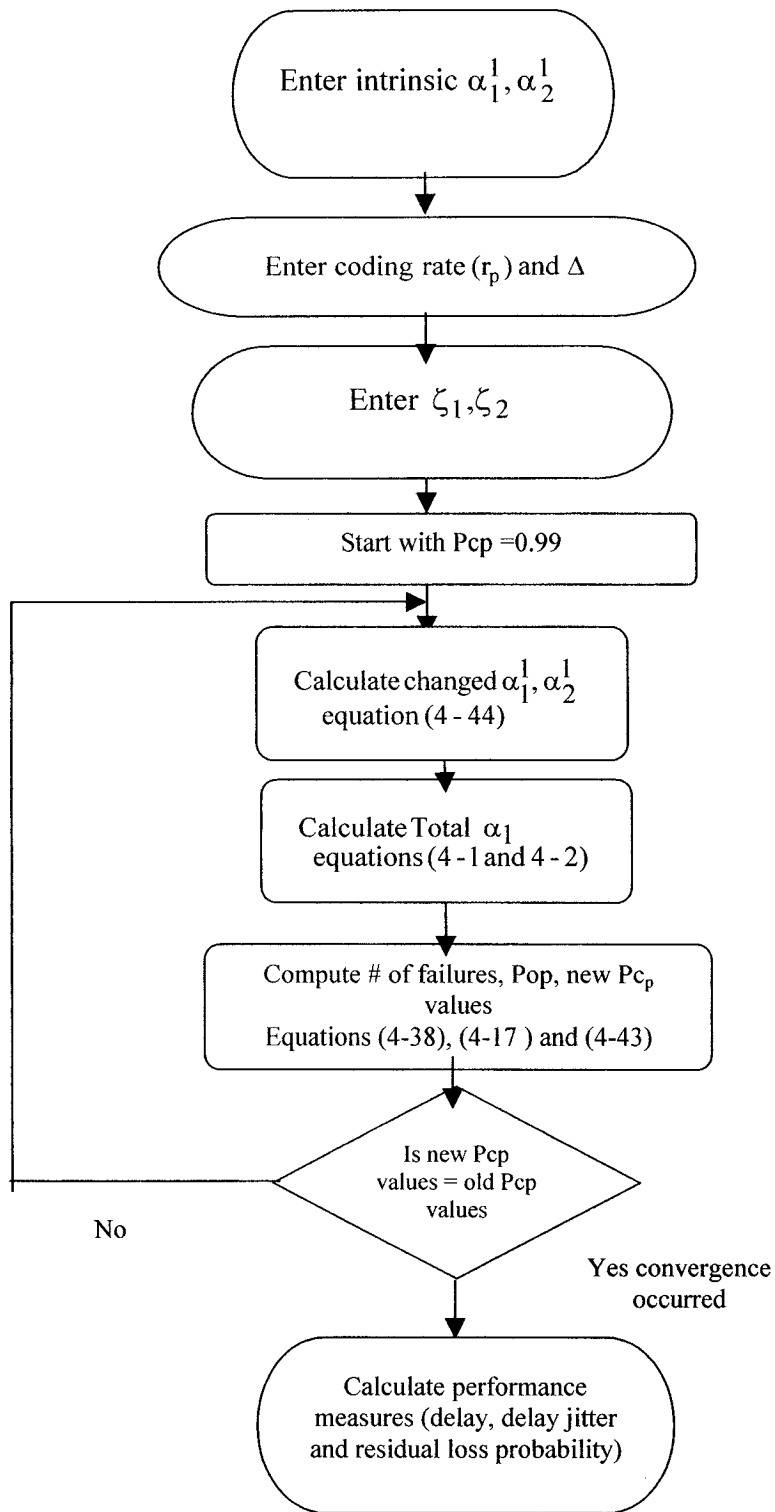


Fig. 4-23 Flowchart of program which calculate performance measures of Hybrid FEC/ARQ unicast repairs

Different input parameters like  $\xi_1, \xi_2, r_p$  and  $\Delta$  are used to calculate the total arrival probabilities  $\alpha_p$  ( $p=1,2,3$ ). All of these parameters are set by the network designer and are his choice.

Solving the coupled state diagrams in Fig. 4-2 necessitate the substitution of  $P_{c_p}$  values and various traffic probabilities  $\alpha_p^1$  into the coupled state diagrams of Fig. 4-2. Assume the knowledge of  $P_{c_p}$  and  $\alpha_p^1$  values. However, these values depend on the multicast conditions and the number of retransmission trials ( $T_p$ ), etc. We have assumed a suitable starting value of  $P_{c_p} = 0.99$  and the values of intrinsic arrivals as in the analysis figures of this chapter; solve the buffer state equations thus obtaining  $P_{o_p}$  ( $P_{e_p}$  is given a value say  $10^{-7}$ , the designer may change this one depending on network condition) as in equation 4-43. Equation 4-43 then yields the new  $P_{c_p}$  values which should be substituted instead of old  $P_{c_p}$  values in state diagrams of Fig. 4-2. Also, the total number of failures (from equation 4-38) lead to magnifying the values of  $\alpha_p^1$  as in equation 4-44. These new magnified values should be substituted back in place of the initial values of  $\alpha_p^1$  in Fig. 4-2, and so on till to converge in the values of  $P_{c_p}$ . After convergence, performance measures like total packet delay, total packet delay jitter and the residual packet loss probability are calculated. See Flowchart of Fig. 4-23.

### 4.3.3 Analysis Results

Figs. 4-24 to 4-32 show the performance comparisons between IP sources and MPLS sources in the multicast tree when neither FEC nor ARQ is applied. Fig. 4-24 shows the total packet delay for all sources for both IP and MPLS versus IP factor for small processing factor ( $\tau$ ). It shows that IP and MPLS have almost the same total packet delay, except a small difference for source 3. Fig. 4-25 shows the total delay jitter for all sources for both IP and MPLS versus IP factor for small processing factor. Also, it shows that both IP and MPLS have very much the same total delay jitter, except a small difference for source 3; and as IP factor increases the difference becomes even smaller. Fig. 4-26 shows the residual packet loss probability for all sources for both IP and MPLS versus IP factor for small processing factor. Also, it shows that IP and MPLS sources have very same residual loss probability (almost zero), except for IP source 3.

However, Figs. 4-27, 4-28 and 4-29 show that when the processing factor ( $\tau$ ) increases MPLS will have superiority over IP in terms of the total packet delay, total delay jitter and the residual packet loss probability. As shown in Figs. 4-27 and 4-29, the total packet delay and the residual packet loss probability in case of MPLS are less than IP for all sources and these differences are clear for low priority sources 2 and 3. Fig. 4-28 shows that the total delay jitter in the case of MPLS is less than IP for all sources except for MPLS source 3 which starts smaller than the IP source 3 and it continues to increase with the increase of IP factor. This means when the difference in packet processing ( $\tau$ ) between MPLS and IP increases, MPLS in general will be better.

In Figs. 4-24 to 4-29 MPLS factor was constant and relatively small; explaining why MPLS performance was better or very similar to IP performance. However, in the

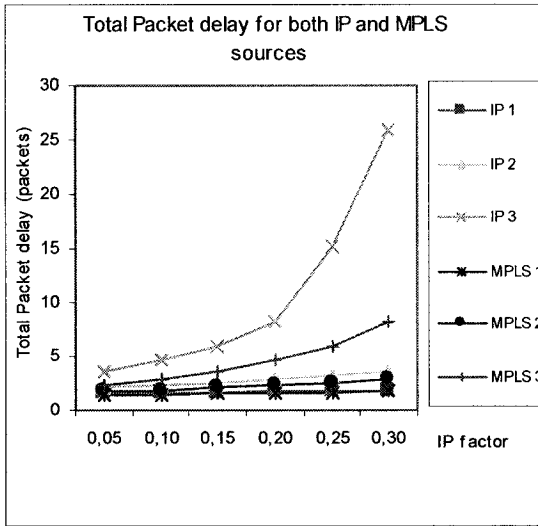
following figures we will study the effects of MPLS factor on MPLS performance. Figs. 4-30, 4-31 and 4-32 show that IP will be superior over MPLS when MPLS factor increases. As shown in Figs. 4-30, 4-31 and 4-32 the total packet delay, the total delay jitter and the residual packet loss probability in the case of IP (which are constant) are slightly less than MPLS. This means when the extra arrival rate due MPLS control overhead used to establish MPLS multicast paths or tree increases, IP will be performing better especially when the intrinsic traffics increase.

Figs. 4-33 to 4-36 consider the performance comparisons between IP sources and MPLS sources in the multicast tree when FEC mechanism only is applied. The tendencies of Figs. 4-33 to 4-36 are similar to the case of Figs. 4-27 to 4-30. However, when using FEC there would be a slight increase in the total packet delay for all IP and MPLS sources compared to without using FEC or ARQ due to the increase in intrinsic arrival probabilities because of the FEC operation. However, the residual packet loss probability for all sources would decrease due to the use of improved  $P_c$  value in the case of FEC only.

Figs. 4-37 to 4-40 consider the performance comparisons between IP sources and MPLS sources in the multicast tree when ARQ multicast mechanism is applied. The tendencies of Figs. 4-37 to 4-40 are very similar to the case of Figs. 4-27, 4-28, 4-29 and 4-31. However, the superiority of MPLS sources over IP sources in terms of total packet delay, total delay jitter and the residual packet loss probability is clearer in Figs. 4-37, 4-38 and 4-39 when the processing factor is large. Also, the superiority of IP sources over MPLS sources in terms of total delay jitter is clearer in Fig. 4-40 when the MPLS factor is large.

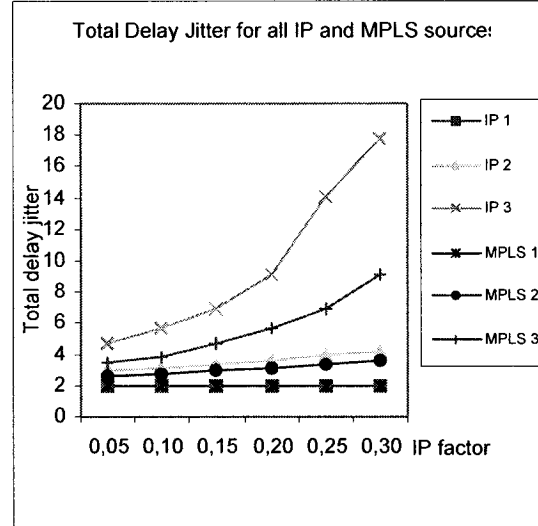
Figs. 4-41 to 4-44 consider the performance comparisons between IP sources and MPLS sources in the multicast tree when ARQ unicast mechanism is applied. The tendencies of Figs. 4-41 to 4-44 are very similar to the case of Figs. 4-37 to 4-40 when ARQ multicast is used. Using ARQ only would have the worst total packet delay for all IP and MPLS sources compared to without FEC or ARQ (case 1) or FEC only (case 2) due to retransmission request. However, using ARQ only would improve the residual packet loss probability in a noticeable manner. ARQ unicast would be better than ARQ multicast in terms of residual packet loss probability but worst than it in terms of total packet delay.

Figs. 4-45 to 4-52 show the performance comparisons between IP sources and MPLS sources in the multicast tree when hybrid FEC/ARQ mechanism is applied. The tendencies of these figures are very similar to the case of Figs. 4-37 to 4-44. Using hybrid FEC/ARQ would have the best residual packet loss probability among all schemes and the worst total packet delay among all schemes. In addition to that the hybrid FEC/ARQ unicast subcase performs better than hybrid FEC/ARQ multicast in terms of residual packet loss probability but worst than it in terms of total packet delay.



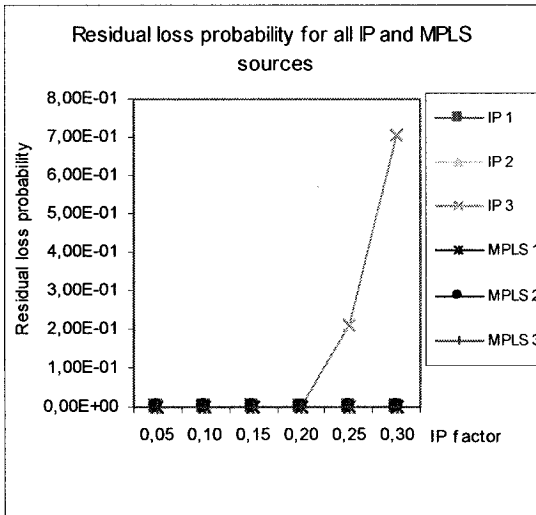
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2, L = 500$

Fig. 4-24 Total packet delay versus IP factor (No FEC or ARQ and small  $\tau$ )



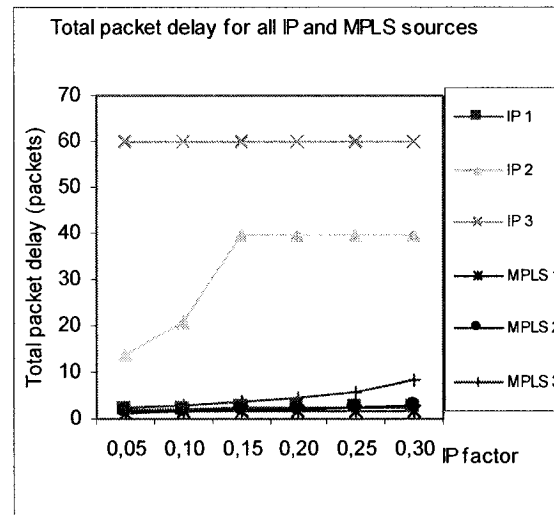
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2, L = 500$

Fig. 4-25 Total delay jitter versus IP factor (No FEC or ARQ and small  $\tau$ )



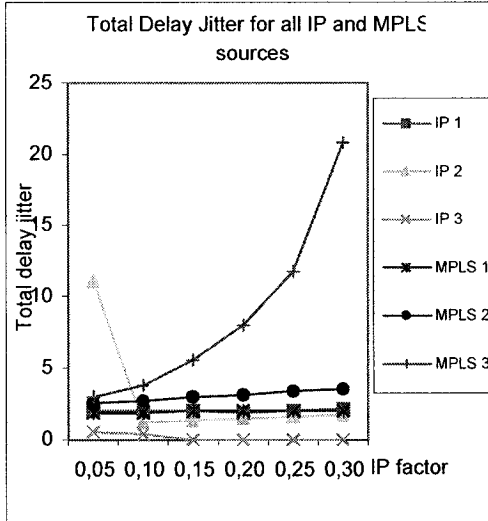
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2, L = 500$

Fig. 4-26 Residual loss probability versus IP factor (No FEC or ARQ and small  $\tau$ )



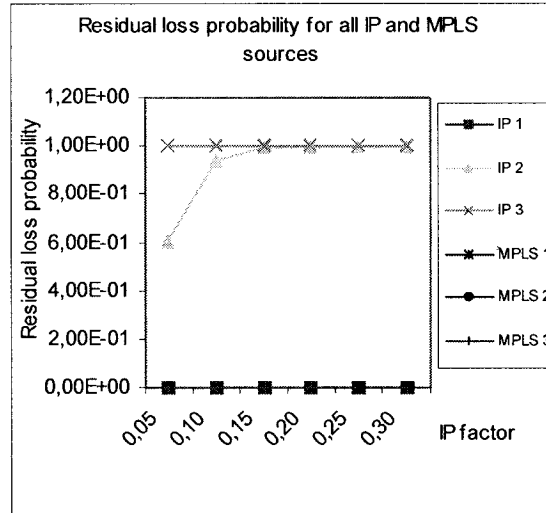
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-27 Total packet delay versus IP factor (No FEC or ARQ and large  $\tau$ )



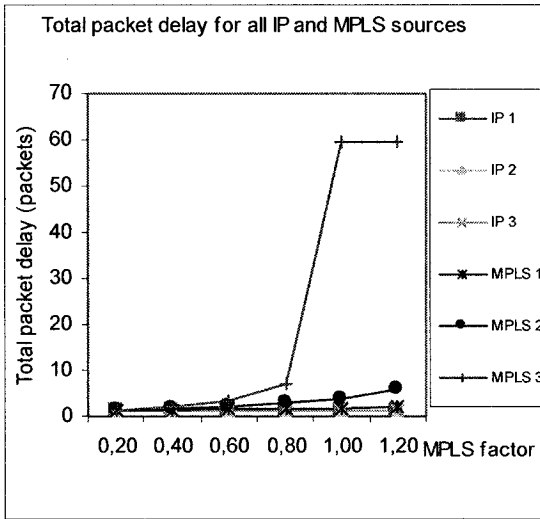
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-28 Total delay jitter versus IP factor (No FEC or ARQ and large  $\tau$ )



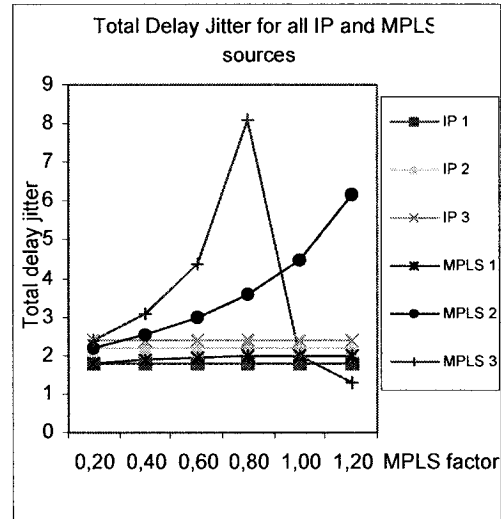
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-29 Residual loss probability versus IP factor (No FEC or ARQ large  $\tau$ )



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.10, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$

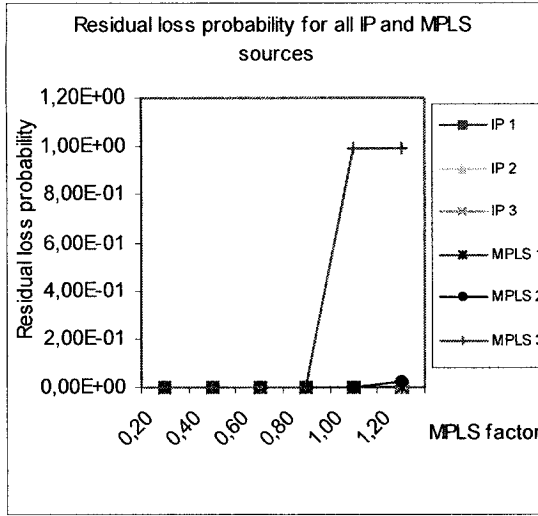
Fig. 4-30 Total packet delay versus MPLS factor (No FEC or ARQ)



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.10, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$

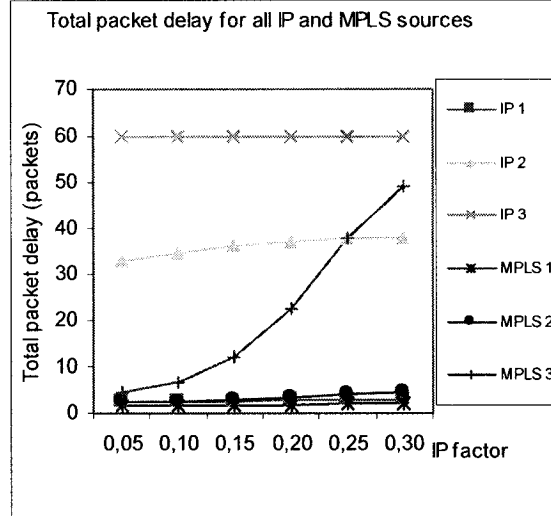
Fig. 4-31 Total delay jitter versus MPLS factor (No FEC or ARQ)





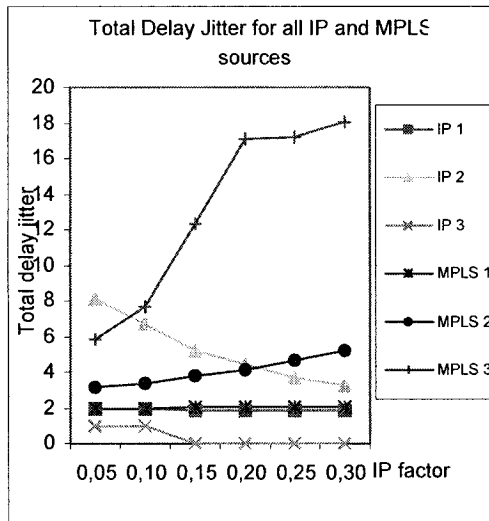
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.10, \beta_1 = 1,$   
 $D = 4, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$

Fig. 4-32 Residual loss probability versus MPLS factor (No FEC or ARQ)



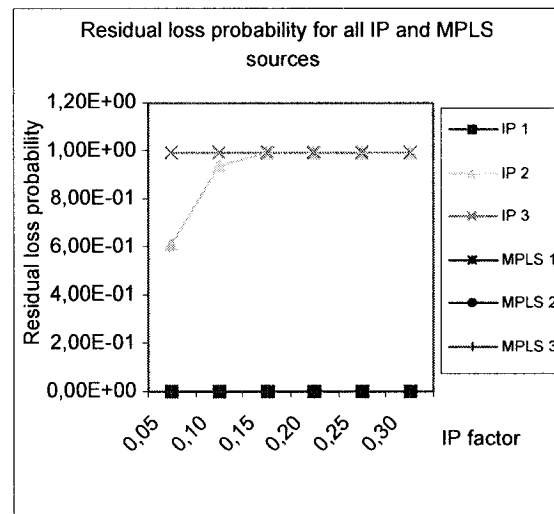
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1, L = 500$   
 $r = 223/255, D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$

Fig. 4-33 Total packet delay versus IP factor (FEC only and large  $\tau$ )



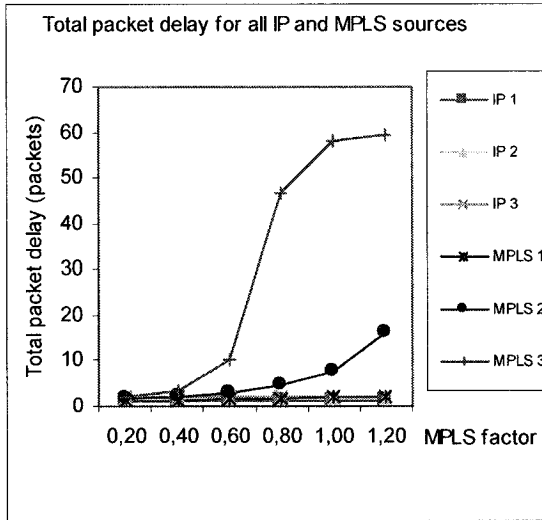
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1, L = 500$   
 $r = 223/255, D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$

Fig. 4-34 Total delay jitter versus IP factor (FEC only and large  $\tau$ )



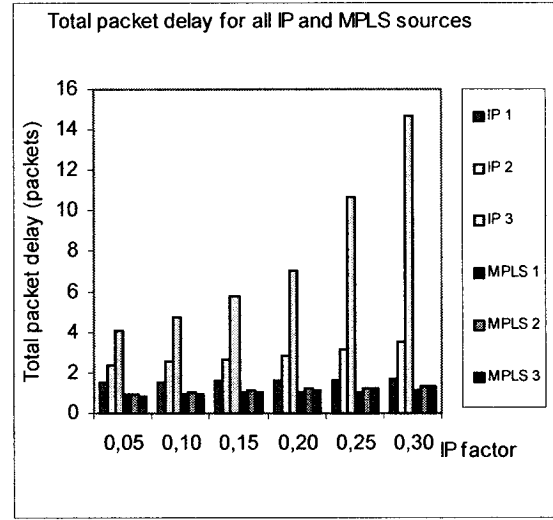
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \beta_1 = 1, L = 500$   
 $r = 223/255, D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$

Fig. 4-35 Residual loss probability versus IP factor (FEC only and large  $\tau$ )



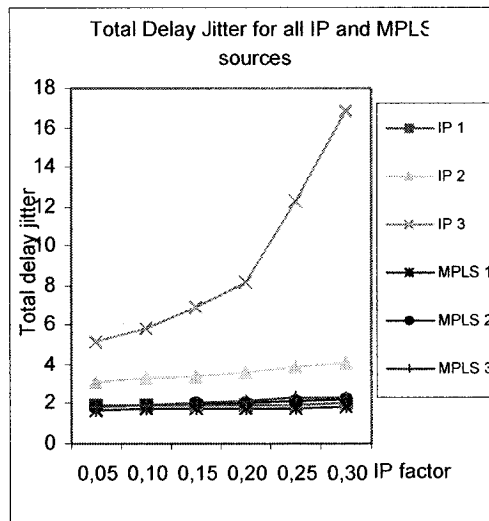
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, L = 500$   
 $r = 223 / 255, D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2$

Fig. 4-36 Total packet delay versus MPLS factor (FEC only)



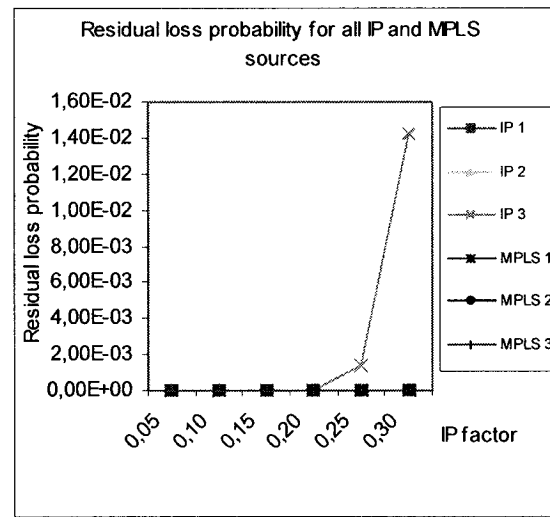
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-37 Total packet delay versus IP factor (ARQ multicast and large  $\tau$ )



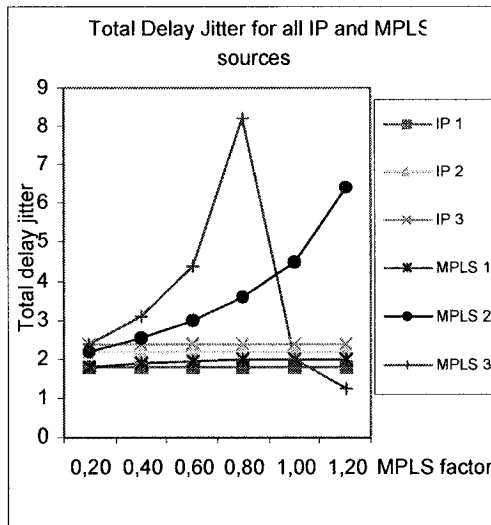
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-38 Total delay jitter versus IP factor (ARQ multicast and large  $\tau$ )



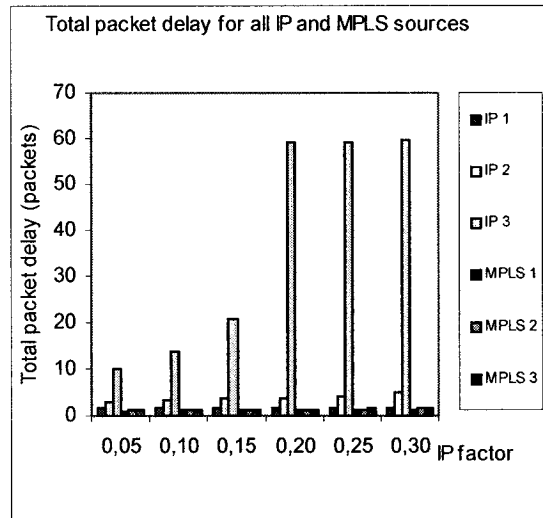
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500$

Fig. 4-39 Residual loss probability versus IP factor (ARQ multicast and large  $\tau$ )



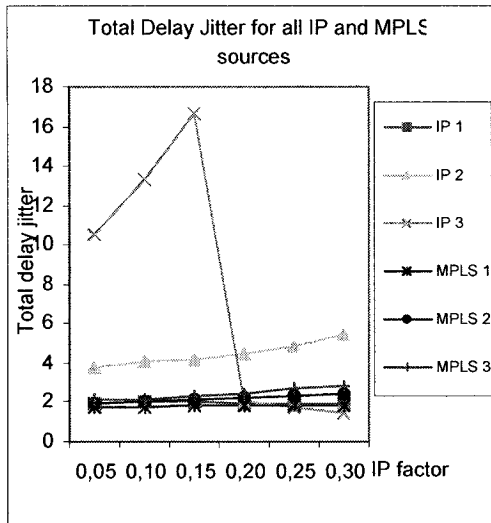
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.10, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$

Fig. 4-40 Total delay jitter versus MPLS factor (ARQ multicast)



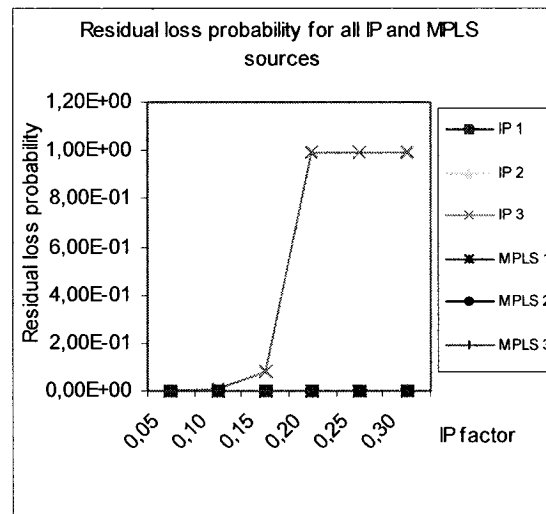
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, \Delta = 0.1$

Fig. 4-41 Total packet delay versus IP factor (ARQ unicast and large  $\tau$ )



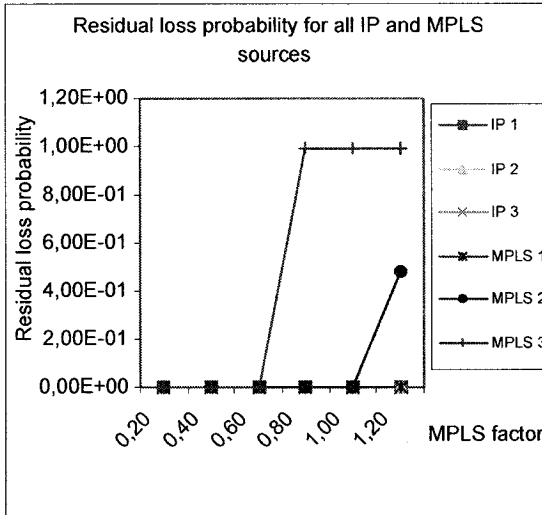
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, \Delta = 0.1$

Fig. 4-42 Total delay jitter versus IP factor (ARQ unicast and large  $\tau$ )



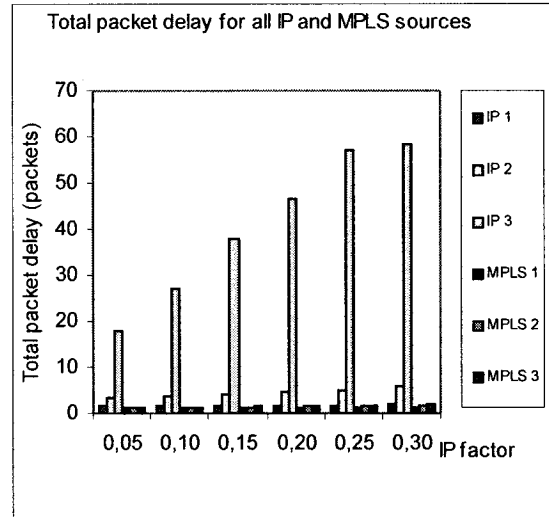
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, \Delta = 0.1$

Fig. 4-43 Residual loss probability versus IP factor (ARQ unicast and large  $\tau$ )



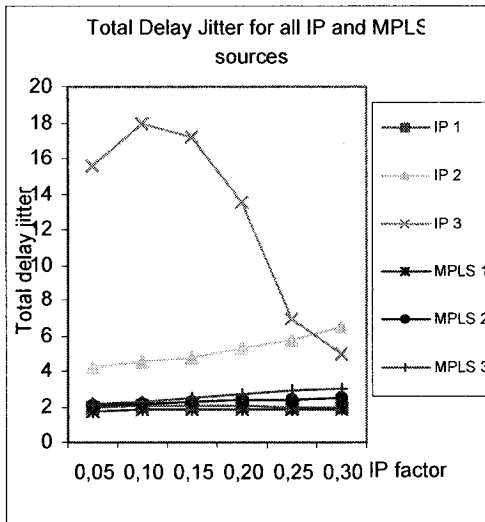
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.10, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500, \Delta = 0.1$

Fig. 4-44 Residual loss probability versus MPLS factor (ARQ unicast)



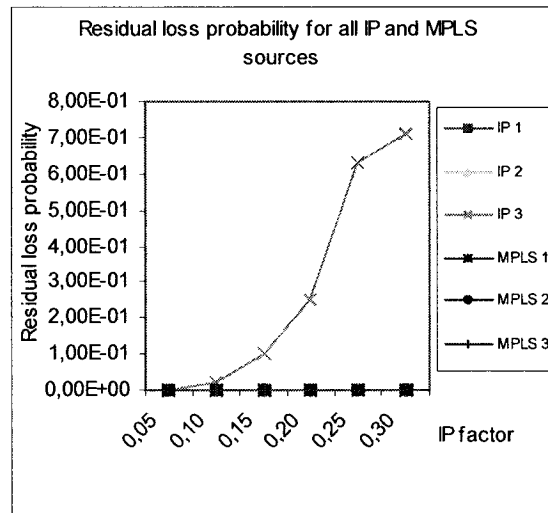
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/254$

Fig. 4-45 Total packet delay versus IP factor (Hybrid multicast and large  $\tau$ )



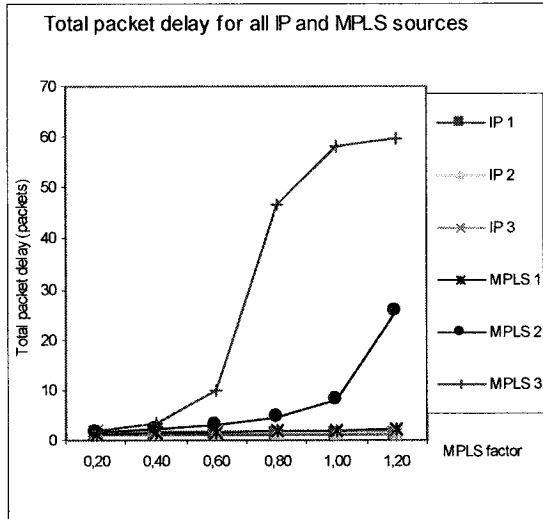
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/254$

Fig. 4-46 Total delay jitter versus IP factor (Hybrid multicast and large  $\tau$ )



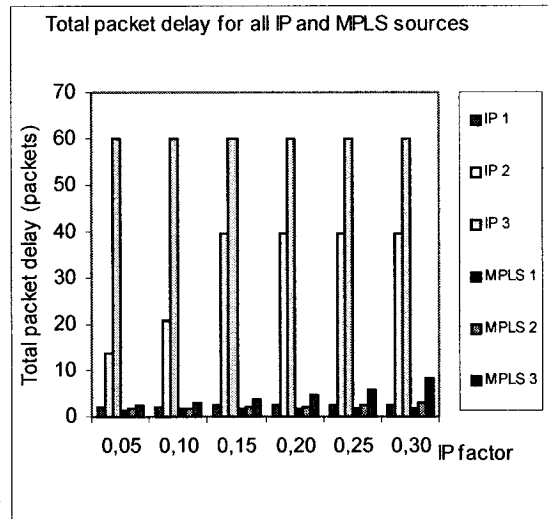
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/254$

Fig. 4-47 Residual loss probability versus IP factor (Hybrid multicast and large  $\tau$ )



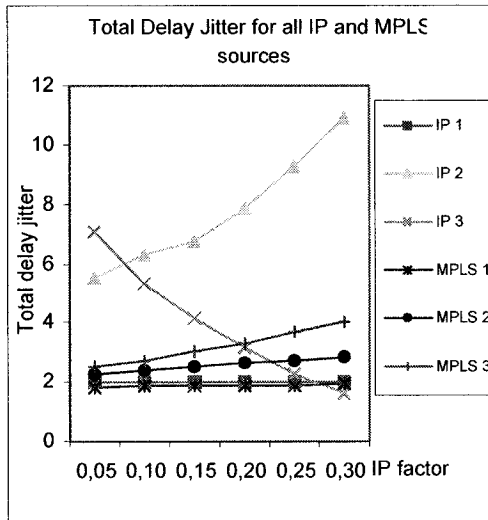
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2$   
 $D = 4, B = 30, \xi_1 = 0.1, \tau = 1.2, L = 500, r = 223/255$

Fig. 4-48 Total packet delay versus MPLS factor (Hybrid multicast)



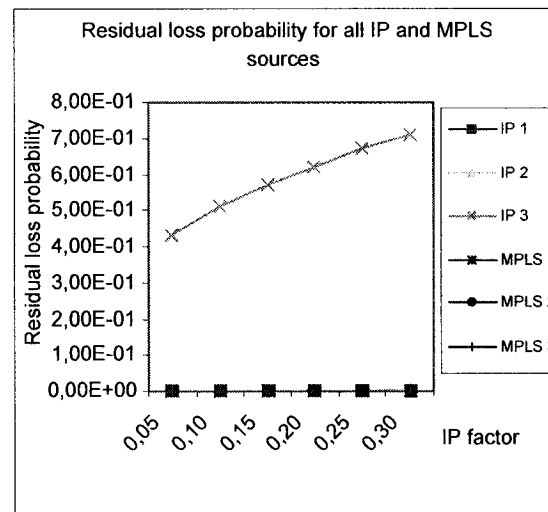
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2, \Delta = 0.1$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/255$

Fig. 4-49 Total packet delay versus IP factor (Hybrid unicast and large  $\tau$ )



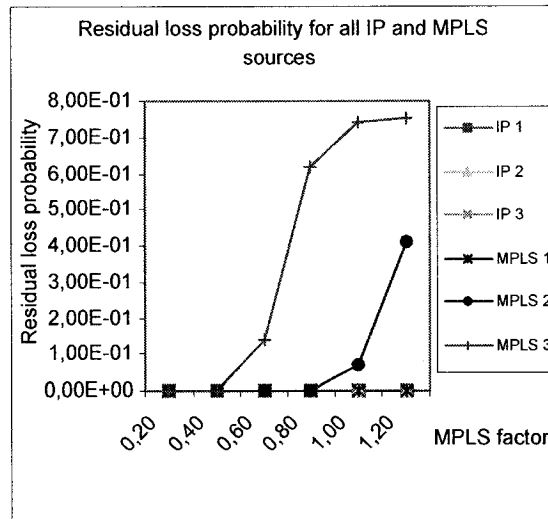
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2, \Delta = 0.1$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/255$

Fig. 4-50 Total delay jitter versus IP factor (Hybrid unicast and large  $\tau$ )



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2, \Delta = 0.1$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8, L = 500, r = 223/255$

Fig. 4-51 Residual loss probability versus IP factor (Hybrid unicast and large  $\tau$ )



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \beta_1 = 1, z = 2, \Delta = 0.1$$

$$D = 4, B = 30, \xi_1 = 0.1, \tau = 1.2, L = 500, r = 223/255$$

Fig. 4-52 Residual loss probability versus MPLS factor (Hybrid unicast)

## 4.4 Conclusions

In this chapter, a performance comparison between IP multicast trees and MPLS multicast trees is carried using analysis tools. In addition to that a new Fair Share Policy (FSP), which is a traffic policing mechanism, is proposed to ensure proper QoS. Also, Differentiated Services and reliable multicasting are used in this comparison. We found that when the difference in packet processing time ( $\tau$ ) between IP and MPLS is high and when MPLS factor is small, IP multicast will perform less efficiently than MPLS in terms of the total packet delay, total delay jitter and the residual packet loss probability. However, when this difference in packet processing time is small IP performs very similar to MPLS. In addition to that when MPLS has higher arrival rate due to MPLS

trees establishment control overhead and when the processing factor is small, IP would perform better than MPLS.

Analysis results revealed that there is a noticeable improvement in QoS defined as the total packet delay, total delay jitter and the residual packet loss probability for a higher priority traffic when MPLS multicasting replaces IP multicasting especially if MPLS factor is small and processing factor is large. However, the difference between the two QoS provided by MPLS and IP becomes minimal for low priority traffic.

In addition to that, the study finds that the no FEC or ARQ mechanism (case1) is the best mechanism in terms of total packet delay for all IP and MPLS sources, and the hybrid FEC/ARQ unicast mechanism is the best mechanism in terms of the residual packet loss probability for all IP and MPLS sources.

The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). Each router may have different capabilities; for example one router could have the ability to correct errors (FEC) and use ARQ, one may use only ARQ but cannot correct errors, a third one may not have MPLS capability. In this chapter, the study carried only homogeneous networks. In chapter 6, heterogeneous networks would be considered.

# CHAPTER 5 RESIDUAL PACKET LOSS PROBABILITY FOR DIFFSERV OVER IP AND MPLS MULTICAST TREES

## 5.1 Introduction

In order to achieve a better quality of service (QoS), the use of reliable multicasting has become increasingly important especially with the emergence of Internet-based applications such as IP telephony and audio/video conferencing. The ARQ Multicast repairs mechanism (considered herein) is simpler than the ARQ unicast repairs mechanism and requires less overhead; however the multicast repairs mechanism consumes much more bandwidth.

In this chapter, the worst case residual packet loss probability in a complete binary multicast tree which consists of  $N$  routers is evaluated for both IP and MPLS multicast trees, when Automatic Repeat Request (ARQ) with multicast repairs mechanism is employed and when DiffServ is adopted. We also derive and compare two other mathematical expressions, which can be used to calculate the residual packet loss probability in binary multicast trees for both IP and MPLS. The first expression deals only with the number of routers that should have correct transmissions (no loss and no errors) during the repair trial of one previous loss depending on the location of this previous loss. The second one takes into account the number of trials, the number of errors and the position of each error (at which level of the multicast tree, the error occurred).



In [126-127], the residual packet loss probability in a complete binary multicast tree which consists of  $N$  routers with a given probability of successful delivery to the next router is evaluated when Automatic Repeat Request (ARQ) with multicast repairs is employed.

In this chapter, we present the same fair share policy (FSP) with same discrete coupled state diagrams [chapter 4] for accommodating priority sessions while not degrading much the QoS of low priority sessions. By taking the above constraints into consideration, we evaluate the QoS performance in terms of residual packet loss probability for a typical binary tree in the two cases of IP and MPLS multicasting. We also consider Differentiated Services; i.e. traffics with different priority classes when ARQ with multicast repairs is used. Analysis tools will be used to evaluate the fair share policy (FSP) for different homogeneous network scenarios.

We use a complete binary multicast tree, where each parent router has two children routers until we reach leafs. Fig. 5-1 shows an example of a complete homogeneous binary multicast tree with the root, which is the nearest router or node to the sender or the rendezvous point, and the leafs, which are the routers with receivers underneath them. As shown in the figure the depth of this tree is 5 and the total number of routers  $N$  is 31. This tree could represent either an MPLS multicast tree or an IP multicast tree.

The chapter is organized as follows. In section 5.2, two methods for calculating the residual packet loss probability in case of IP or MPLS multicast are introduced, namely: worst case and approximate methods. In section 5.3, a more exact method is formulated to calculate the residual packet loss probability. In section 5.4, comparative analysis results of the three different methods, used to calculate the residual packet loss

probability, in the two cases of IP and MPLS multicasting are discussed. Finally section 5.5 includes a summary of the conclusions.

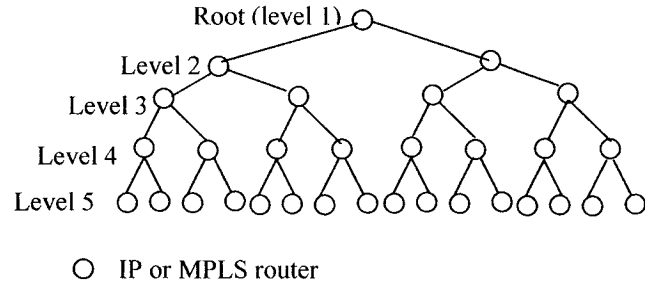


Fig. 5-1 A complete homogeneous binary multicast tree

## 5.2 Residual Packet Loss Probability Calculations Using Worst Case and Approximate Methods

In this section, two methods for calculating the residual packet loss probability in case of IP or MPLS multicast are formulated, namely: worst case and approximate methods.

Upon the receipt of a NAK from one or more receivers, the sender multicast again the repair packet to all receivers. The probability of success of one trial for worst case scenario [126-127] for priority traffic p is given as:

$$P_{s_{pw}} = P_{c_p}^N \quad (\text{worst case scenario}) \quad (5-1)$$

Therefore, the residual packet loss probability of one trial for worst case scenario [126-127] for priority traffic p is given as:

$$P_{loss_{pw}} = 1 - P_{s_{pw}} \quad (\text{worst case scenario}) \quad (5-2)$$

Where  $P_{c_p}$  is the probability of successful delivery to next router for priority traffic p and N is the number of routers in the binary multicast tree.  $P_{c_p}$  is given as:

$$P_{c_p} = (1 - P_{o_p} - P_{e_p})^L \quad (5-3)$$

Where  $P_{o_p}$  is the byte overflow (loss) for priority traffic  $p$  which can be given as:

$$P_{o_p} = \frac{\text{Packet Loss (Overflow) Probability}}{\text{Number of bytes per packet}} = \frac{P_{Lp}}{L} \quad (5-4)$$

$P_{e_p}$  is the byte error probability for certain priority traffic  $p$  and  $L$  is the number of bytes per packet. In the previous equation, two assumptions are made:

- 1- Loss of source packet is caused by consecutive byte losses at the intermediate routers.
- 2- Interleaving is used in order to break byte burst losses and efficiently turn them into independent random byte losses at the source and destination [125].

Equations 5-1 and 5-2 represent upper bounds for the probability of success and the residual packet loss probability in the worst case scenario when ARQ with multicast repairs mechanism is used. However, using ARQ multicast repairs will have a better chance of success with each trial since the number of receivers who did not receive the packet correctly decreases with each trial. Therefore, the average probability of success in a typical transmission multicast trial for priority traffic  $p$  can be approximated as:

$$P_{S_{pA}} = \frac{Pc_p^N + Pc_p^{(N/2)+1} + Pc_p^{(N/4)+2} + Pc_p^{(N/8)+3} + Pc_p^{(N/16)+4} + \dots + Pc_p^{(N/2^D)+D-1}}{D \text{ (number of tree levels)}} \quad (5-5)$$

Where  $D$  is the network depth. If the packet does not suffer loss or error on any of the  $N$  routers of the multicast tree, with probability  $Pc_p^N$  no further repair is needed, this explains the first term of equation 5-5. However, if there has been an error or loss which is located at level 1 (see Fig. 5-1), then the repair packet would be sent from sender to the router at level 1, and then the repair packet will flow to  $N/2$  routers under level 1. All such  $(N/2) + 1$  transmissions of repair packet have to be correct, otherwise further repair is needed and so on. The probability of these  $(N/2) + 1$  correct transmissions of subject repair packet is given by  $Pc_p^{(N/2)+1}$  and so on for the remaining terms in equation 5-5.

We divide by  $D$  (network depth) because we assume that errors are equally likely to occur on different levels of the tree giving rise to the addition of different terms (equation 5-5) and the division by the depth  $D$  where  $D = \log_2(N+1)$ .

The total number of ARQ trials for priority  $p$  traffic  $T_p$  can be averaged as:

$$T_p = P_{s_p} + 2P_{s_p}(1 - P_{s_p}) + 3P_{s_p}(1 - P_{s_p})^2 + \dots \quad (5-6)$$

Therefore, the number of failures (or retransmissions only) for certain traffic priority  $p$  can be given as an average of a geometrically distributed random variable, i.e.:

$$F_p = T_p - 1 \quad (5-7)$$

Where  $P_{s_p}$  is the average (approximate) probability of packet multicast success for priority traffic  $p$  corresponding to one ARQ trial.  $F_p$  can be easily written in a closed form as:

$$\left\{ \begin{array}{l} F_p = \frac{1}{P_{s_p}} - 1 \text{ for infinite number of ARQ trials} \\ \text{and } F_p = \frac{1 - (z+1)(1 - P_{s_p})^z + (z)(1 - P_{s_p})^{z+1}}{P_{s_p}} - 1 \text{ for } (z) \text{ trials} \end{array} \right. \quad (5-8)$$

Defining  $P_{s_p}'$  as the final approximate probability of success for certain priority traffic  $p$ :

$$\begin{aligned} P_{s_p}' &= P_{s_p} + (1 - P_{s_p})P_{s_p} + (1 - P_{s_p})^2 P_{s_p} + \dots + P_{s_p}(1 - P_{s_p})^{z-1} \\ &= 1 - (1 - P_{s_p})^z \quad \text{for } (z) \text{ trials of a typical packet to the multicast tree.} \end{aligned} \quad (5-9)$$

Where we note that for one trial  $P_{s_p}' = P_{s_p}$  and  $P_{s_p}' = 1$  for infinite retransmission trials as should be. Therefore, the approximate residual loss probability (after all ARQ trials) is given by:

$$P_{loss_{pA}} = 1 - P_{s_p}' \quad (5-10)$$

Due to the use of ARQ multicast repairs, the intrinsic arrival probability  $\alpha_p^1$  for certain priority traffic  $p$  would increase according to:

$$\alpha_p^1 = \alpha_p^1(1 + F_p), \quad p = 1, 2, 3 \quad (5-11)$$

### 5.3 Residual Packet Loss Probability Calculation Using an Exact Method

In this section, a more exact method is formulated to calculate the residual packet loss probability. Fig. 5-2 explains the tree of successful repair of 1 error in 4 trials (or 3 retransmissions) for priority  $p$  traffic. The figure shows that the loss or error can occur at any level (2, 3, ...  $D$ ) where  $D$  is the multicast tree depth. In each trial  $z$  ( $z=1, 2, 3$  or 4), there could be either a success (S) or a failure (F) [126-127].

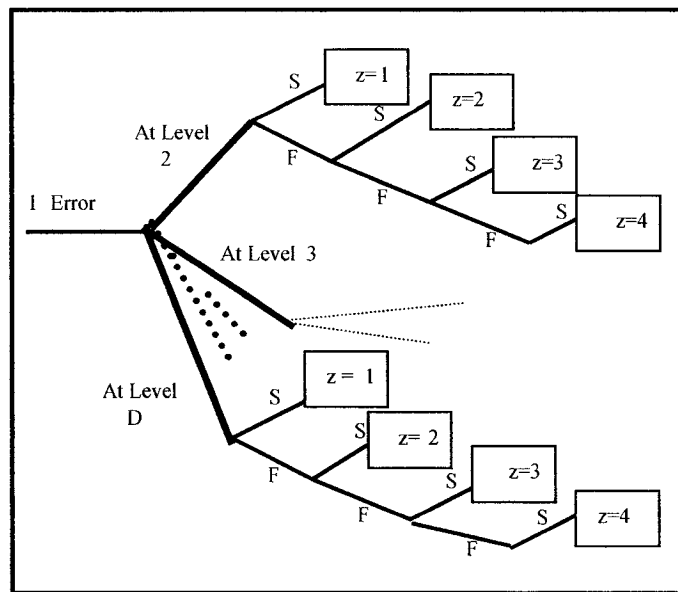


Fig. 5-2 The 4 trials tree of success to repair 1 error for certain priority traffic  $p$

Defining  $P_{s_p}'$  as the final exact probability of success for priority traffic p:

$$P_{s_p}' = \omega_{1p} + \omega_{2p} + \omega_{3p} + \omega_{4p} \quad (5-12)$$

Where  $\omega_{ip}$  represents the probability of success in each trial ( $i=1,2,3,4$ ) for priority traffic p. Where:

$$\omega_{1p} = P_{c_p}^N \quad (5-13)$$

Equation 5-13 means that if all routers have received the packet correctly from the first trial we stop sending the same packet. However, if an error happens in the first trial and is being corrected in the second trial. Then the probability of success of same packet in the second trial, i.e. all remaining users who did not get the packet in the first trial will get it in the second one, will be given as:

$$\omega_{2p} = \sum_{j=2}^D \binom{N}{1} * (2^{j-1} / N) * (1 - P_{c_p}) P_{c_p}^{N-1} P_{c_p}^{(N/2^{j-1})+(j-1)} \quad (5-14)$$

Equation 5-14 shows that any of the N routers could not receive the packet which explains the first term  $\binom{N}{1}$  and this 1 error could have happened at any level with probability  $(2^{j-1} / N)$ . The term  $(1 - P_{c_p}) P_{c_p}^{N-1} P_{c_p}^{(N/2^{j-1})+(j-1)}$  explains that there was 1 error and was not corrected in the first trial with probability  $(1 - P_{c_p})$  and all the routers have received the packet except 1 router with probability  $(P_{c_p}^{N-1})$  and this error was corrected in the second trial with probability  $P_{c_p}^{(N/2^{j-1})+(j-1)}$  which says that all  $(j-1)$  routers which are the preceding routers (fathers) of the router that experienced the error have received the packet correctly and all children routers that are underneath this router,  $(N/2^{j-1})$  of them have received the packet correctly. Similarly equation 5-15 shows that the error was not corrected at the 2<sup>nd</sup> trial with probability  $(1 - P_{c_p}^{(N/2^{j-1})+(j-1)})$  and is corrected

only at the 3<sup>rd</sup> trial. Also, equation 5-16 shows that the probability of unable to correcting the error in the 3<sup>rd</sup> trial was  $(1 - P_{c_p}^{(N/2^{j-1})+(j-1)})$  and the error was actually corrected at the 4<sup>th</sup> trial.

$$\omega_{3p} = \sum_{j=2}^D \binom{N}{1} * (2^{j-1} / N) * (1 - P_{c_p}) P_{c_p}^{N-1} * (1 - P_{c_p}^{(N/2^{j-1})+(j-1)}) * P_{c_p}^{(N/2^{j-1})+(j-1)} \quad (5-15)$$

$$\omega_{4p} = \sum_{j=2}^D \binom{N}{1} * (2^{j-1} / N) * (1 - P_{c_p}) P_{c_p}^{N-1} * (1 - P_{c_p}^{(N/2^{j-1})+(j-1)})^2 * P_{c_p}^{(N/2^{j-1})+(j-1)} \quad (5-16)$$

Therefore, the exact residual loss probability (after all ARQ trials) [126-127] for priority traffic p is given by:

$$P_{loss_{pE}} = 1 - P_{s_p}' \quad (5-17)$$

Solving the coupled state diagrams in Fig. 4-2 necessitate the substitution of  $P_{c_p}$  values and various traffic probabilities  $\alpha_p^1$  into the coupled state diagrams of Fig. 4-2. Assume the knowledge of  $P_{c_p}$  and  $\alpha_p^1$  values. However, these values depend on the multicast conditions and the number of retransmission trials, ... etc. We have assumed a suitable starting value of  $P_{c_p} = 0.99$  and the values of intrinsic arrivals as in figures of this chapter; solve the buffer state equations thus obtaining  $P_{o_p}$  ( $P_{e_p}$  is assumed to be equal to  $10^{-9}$ ) as in equation 5-4. Equation 5-3 then yields the new  $P_{c_p}$  values which should be substituted instead of old  $P_{c_p}$  values in state diagrams of Fig. 4-2. Also, the multicast residual loss probabilities (from equations 5-2, 5-10 and 5-17) lead to magnifying the values of  $\alpha_p^1$  as in equation 5-11. These new magnified values should be substituted back in place of the initial values of  $\alpha_p^1$  in Fig. 4-2, and so on till convergence in the values of  $\alpha_p^1$  and  $P_{c_p}$ .

## 5.4 Analysis Results

Fig. 5-3 shows the comparison between the residual packet loss probabilities for IP source 1 and MPLS source 1 versus IP factor using the worst, approximate and exact calculation methods when number of routers  $N=31$  (Depth=5). IP1W, IP1A, and IP1E are the residual packet loss probabilities for IP source 1 using the worst, approximate and the exact calculation methods. M1W, M1A, and M1E are the residual packet loss probabilities for MPLS source 1 using the worst, approximate and the exact calculation methods. Fig. 5-3 shows that the residual packet loss probability increases with the increase of the intrinsic arrival probability. That's because the increase of the intrinsic arrival probability leads to a decrease in the probability of successful delivery to the next router  $P_c$ . Note that the exact residual packet loss probability in Fig. 5-3 is zero for both IP and MPLS. Figs. 5-4 and 5-5 show similar comparisons but for IP and MPLS sources 2 and 3. Figs. 5-3, 5-4 and 5-5 show that MPLS will have superiority over IP in terms of residual packet loss probability especially when IP and processing factors are large and for low priority traffics. This situation could happen in IP networks using slower subnets, where processing factor at lower layers ( $\tau$ ) is large.

Fig. 5-6 shows the comparison between the residual packet loss probabilities for IP source 2 and MPLS source 2 versus MPLS factor using the worst, approximate and exact calculation methods when number of routers  $N=15$ . The figure shows that IP will have superiority over MPLS in terms of residual packet loss probability when MPLS factor is large and when processing factor is small. Figs. 5-7 and 5-8 show the comparison between the residual packet loss probabilities for IP and MPLS sources 1 and 2



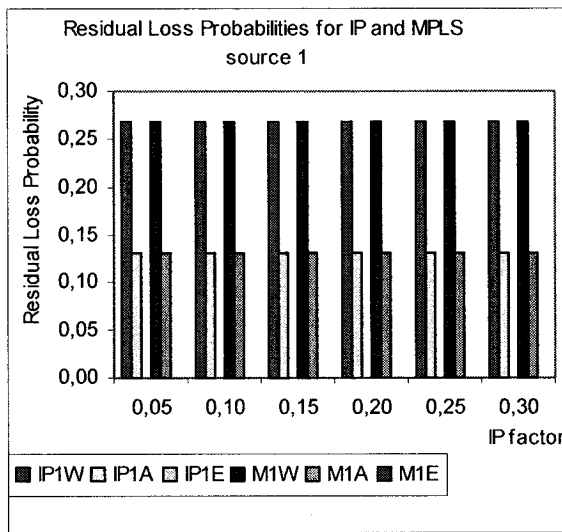
respectively versus MPLS factor using the worst, approximate and exact calculation methods when number of routers  $N=15$ . This comparison is for a lower intrinsic arrival probability than the one in Fig. 5-6. Fig. 5-7 shows that MPLS source 1 and IP source 1 have the same residual packet loss probabilities (very small values) since source 1 has the highest service priority which leads to less buffer overflow probability which in turn makes  $P_c$  for source 1 high. However, Fig. 5-8 shows that IP will have superiority over MPLS in terms of residual packet loss probability when MPLS factor is large and when processing factor is small. This situation could happen in networks that require more time and control overhead to establish and maintain the MPLS multicast tree or paths which will have a direct influence on the increase of MPLS factor.

Figs. 5-9 and 5-10 show similar comparisons to Figs. 5-3 and 5-5 but when  $N=7$ . Fig. 5-9 shows that MPLS source 1 and IP source 1 have the same residual packet loss probabilities (very small values). Similarly Fig. 5-10 shows that MPLS source 3 and IP source 3 have the same residual packet loss probabilities (almost 1). That's because source 3 has the lowest service priority which leads to more buffer overflow probability which in turn makes  $P_c$  for source 3 very small.

Figs. 5-11, 5-12 and 5-13 show the comparison between the residual packet loss probabilities for IP sources (1,2 and 3 respectively) and MPLS sources (1,2 and 3 respectively) versus IP factor using the worst, approximate and exact calculation methods when number of routers  $N=63$  (Depth=6). These figures show that MPLS will have superiority over IP in terms of residual packet loss probability especially when IP and processing factors are large and for low priority traffics. As mentioned previously, this situation could happen in IP networks using slower subnets, where processing factor at

lower layers ( $\tau$ ) is large. Note that the exact residual packet loss probability in Fig. 5-11 is zero for both IP and MPLS.

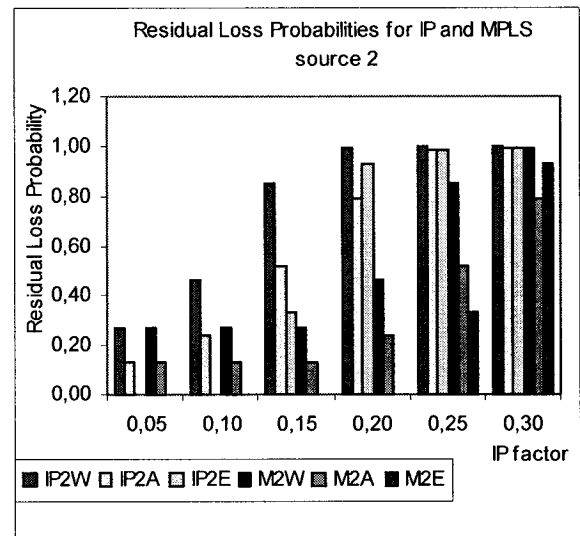
The previous figures show that when the intrinsic arrival probability is high, it will affect the value of  $P_c$  to be small, due to the increase of probability of packet loss (overflow). In this case, there is no big difference between the three methods used to calculate the residual packet loss probability. However, when the intrinsic arrival probability is small and  $P_c$  gets larger, there would be a considerable difference between worst case method and the exact method; in addition to that the approximate method would be very close to the exact method.



$$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$$

$$D = 5, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$$

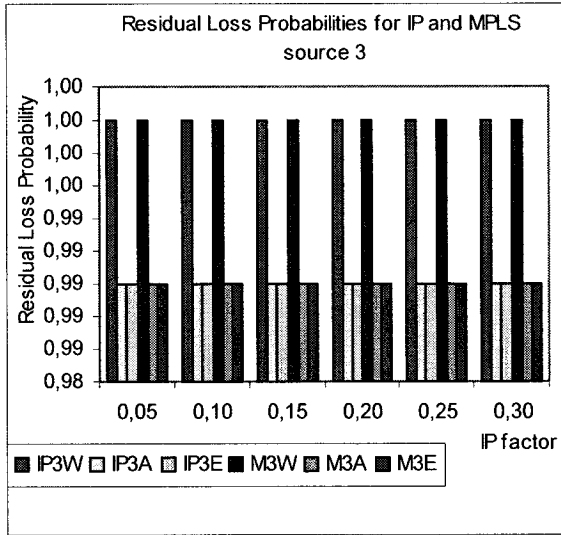
Fig. 5-3 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=31)



$$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$$

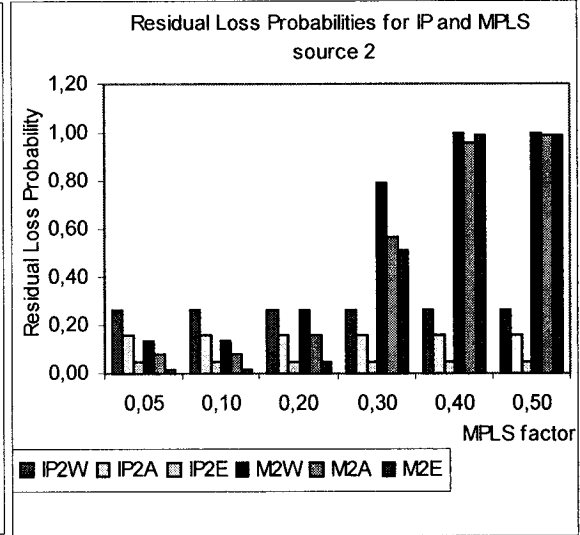
$$D = 5, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$$

Fig. 5-4 Residual Loss Probability for IP and MPLS source 2 versus IP factor (N=31)



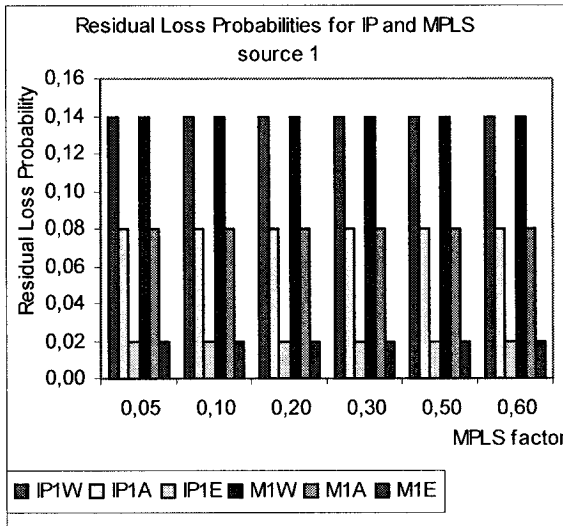
$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 5, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$

Fig. 5-5 Residual Loss Probability for IP and MPLS source 3 versus IP factor (N=31)



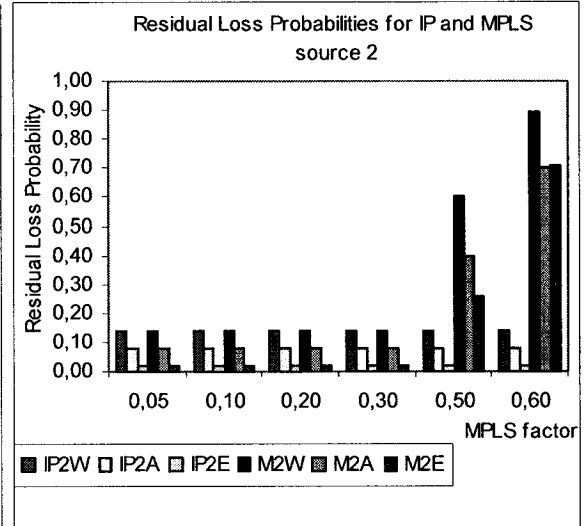
$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 4, B = 30, \xi_1 = 0.15, \tau = 1.2, z = 4$

Fig. 5-6 Residual Loss Probability for IP and MPLS source 2 versus MPLS factor (N=15)



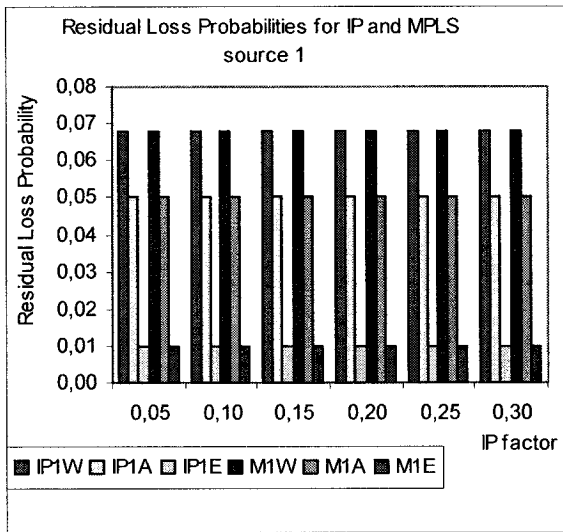
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.25, \alpha_1^3 = 0.2, L = 500$   
 $D = 4, B = 30, \xi_1 = 0.15, \tau = 1.2, z = 4$

Fig. 5-7 Residual Loss Probability for IP and MPLS source 1 versus MPLS factor (N=15)



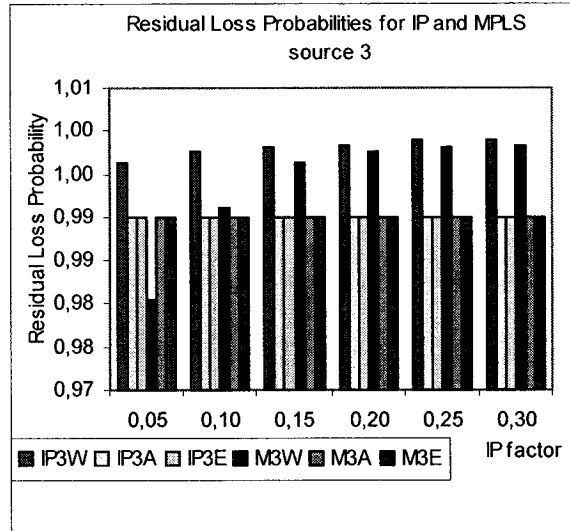
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.25, \alpha_1^3 = 0.2, L = 500$   
 $D = 4, B = 30, \xi_1 = 0.15, \tau = 1.2, z = 4$

Fig. 5-8 Residual Loss Probability for IP and MPLS source 2 versus MPLS factor (N=15)



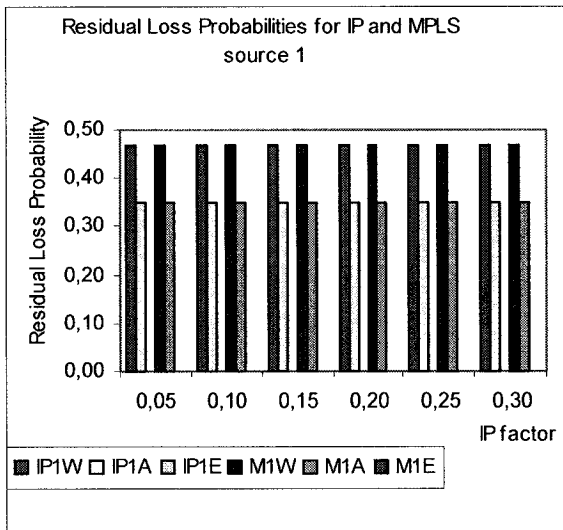
$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 3, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$

Fig. 5-9 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=7)



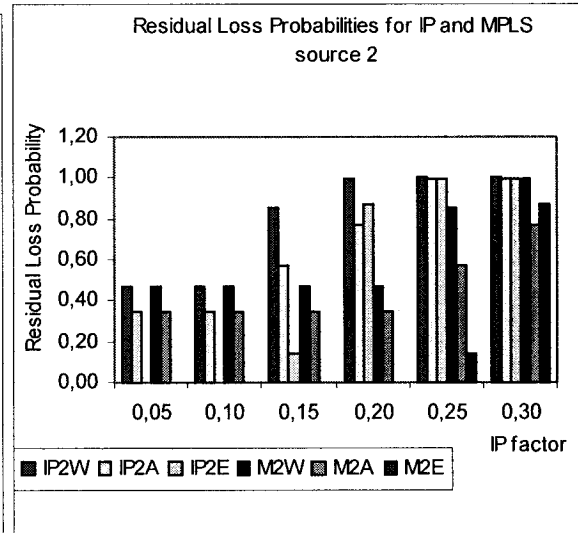
$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 3, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$

Fig. 5-10 Residual Loss Probability for IP and MPLS source 3 versus IP factor (N=7)



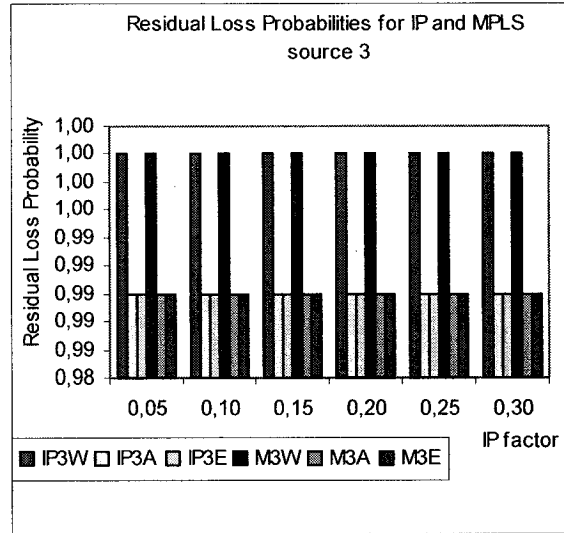
$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 6, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$

Fig. 5-11 Residual Loss Probability for IP and MPLS source 1 versus IP factor (N=63)



$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$   
 $D = 6, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$

Fig. 5-12 Residual Loss Probability for IP and MPLS source 2 versus IP factor (N=63)



$$\alpha_1^1 = 0.35, \alpha_1^2 = 0.3, \alpha_1^3 = 0.25, L = 500$$

$$D = 6, B = 30, \xi_2 = 0.1, \tau = 1.2, z = 4$$

Fig. 5-13 Residual Loss Probability for IP and MPLS source 3 versus IP factor (N=63)

## 5.5 Conclusions

In this chapter, the residual packet loss probabilities in a complete binary IP and MPLS multicast trees which consists of N routers are evaluated for worst case scenario, when Automatic Repeat Request (ARQ) with multicast repairs is employed and when DiffServ is adopted. In addition to that, we have derived and compared two other mathematical expressions, which can be used to calculate the residual packet loss probability in IP and MPLS multicast trees. These expressions are the approximate residual loss probability and the exact residual loss probability. Results show that the approximate residual loss probability is very close and represents a good approximation to the exact value over different ranges of intrinsic arrival probabilities and N values.

Hence, this approximate value can be used to calculate the residual loss probability in case of IP or MPLS binary multicast trees, which will lead to less computational efforts.

Results of this chapter also compare the QoS performance between IP and MPLS multicast trees, when DiffServ and ARQ with multicast repairs are adopted. Results have shown that MPLS will have superiority over IP in terms of residual packet loss probability especially when IP and processing factors are large and for low priority traffics. However, IP will have superiority over MPLS in terms of residual packet loss probability when MPLS factor is large and when processing factor is small.

The chapter presented an approximate but a clear model for the complex DiffServ adaptation where a single interaction between the multicast network and a representative router replaces the  $3N$  dimensional Markovian processes. The later is harder to formulate or solve even for a multicast cluster with  $N=5$  routers.

Though in our analysis, we take the complete binary tree case, there is no loss of generality in taking this full binary tree. The analysis is straightforwardly applicable to non-binary and other types of trees. All one has to do is to replace equations (loss) with non-binary trees counterparts.

The tradeoffs between IP and MPLS multicast networks in the different cases under different traffics are clear from the obtained results.

# CHAPTER 6 RELIABLE QoS MULTICAST FOR DIFFSERV OVER HETEROGENEOUS NETWORKS

## 6.1 Introduction

A number of papers have addressed the problems of QoS in heterogeneous networks [101-112], most of these papers were either descriptive or use simulations only. Also, none of these papers have analyzed the router performance in case of MPLS or IP multicast when reliability and DiffServ are adopted. The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). In this chapter, we define **heterogeneity** as “ **the coexistence of different types of routers with different capabilities in the same network**”. Each router may have different capabilities; for instance one router could be an IP router, a second router could be an MPLS router, a third one could be an MPLS router with FEC capability, and a fourth one could be an IP router with ARQ capability.

In this chapter, we compare the QoS performance in the presence of Fair Share Policy (FSP) of homogeneous IP networks, homogeneous MPLS networks, heterogeneous IP networks and heterogeneous MPLS networks when reliable multicast and DiffServ are used, given their particular constraints.

## 6.2 Reliable QoS Multicast for DiffServ Over Heterogeneous MPLS Networks

In this section, we compare the QoS performance of homogeneous IP networks, homogeneous MPLS networks and heterogeneous MPLS networks when DiffServ and reliable multicasting are used, given their particular constraints. In regular IP multicasting

only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. We present the fair share policy (FSP) and by taking the above constraints into consideration, we evaluate the QoS performance for a typical binary tree in the three mentioned cases. We also consider Differentiated Services; i.e. traffics with different priority classes when reliable multicast is used. Analysis tools will be used to evaluate our fair share policy (FSP).

### **6.2.1 Heterogeneous MPLS networks**

They are three different types of multicast networks. In the homogeneous IP multicast network, all routers are IP routers. In the homogeneous MPLS multicast network, all routers are MPLS routers while in the heterogeneous MPLS multicast network, the network is assumed to be MPLS network but still having some IP routers. This is a practical situation that happens during the migration process from all IP routers to all MPLS routers networking. The number and location of these IP routers in this MPLS network will create the different situations in table 6-1. Each different situation may create up to four types of routers in the MPLS heterogeneous network as would be explained in this section. In this type of network, we can have no IP router in the network (homogeneous MPLS case), 1 IP router in the network, 2 IP routers in the network, or 3 IP routers in the network.

These IP routers can be located anywhere in the network (in our example a complete 31 nodes binary tree is taken) except the root, which is the sender or the Rendezvous Point router. The root is assumed always to be an MPLS router. In any case there will be four types of routers:



- 1- IP (type1) router, which is a regular IP router.
- 2- ME (type 2) router, which is an MPLS router with extra processing due to more packet processing is needed at the MPLS router because the downstream router is an IP router.
- 3- EI (type 3) router, which is either an egress or ingress router with extra processing due to the overhead of tunnel establishment and maintenance and also due to more packet processing is needed because of the IP routers which reside in between EI routers.
- 4- M (type 4) router which is a regular MPLS router.

Fig. 6-1 shows one situation when 1 IP router exists at level 5 (depth) in an MPLS binary network. There is only 1 IP (type 1) router at level 5, 1 ME (type 2) which is an MPLS router with extra processing and there are 29 regular MPLS routers (M or type 4). Note that there are no EI (type 3) routers. Since there are 16 routers at level 5, an IP router can be any one of them. Therefore, there are 16 occurrences of this situation as clarified in table 6-1, and because of the space limit we cannot show them all.

Fig. 6-2 shows another situation when 1 IP router exists at level 4 in an MPLS binary network with depth = 5. There is only 1 IP (type 1) router at level 4; also there are 3 EI (type 3) routers which are Egress and Ingress MPLS routers, 1 of them at the parent level (level 3) and 2 of them are on the children level (level 5). Finally there are 27 regular MPLS routers (M or type 4). Since there are 8 routers at level 4, an IP router can be any one of them. Therefore, there are 8 occurrences of this situation. Also, there are 4 routers at level 3 and any one of them can be an IP router with 3 EI routers and 27 regular MPLS routers, which will give a rise to the number of occurrences of 4. Similarly, there are 2 routers at level 2 and any one of them can be an IP router with 3 EI routers and 27 regular MPLS routers, which will give a rise to the number of occurrences of 2. Therefore, by

summing the total number of occurrences of this kind of situation which is 1 IP router, 3 EI routers and 27 regular MPLS routers, the number would be  $8 + 4 + 2 = 14$  occurrences as shown in table 6-1. Table 6-1 shows the distributions of all existence possibilities of 1 IP, 2 IP or 3 IP routers in a heterogeneous MPLS network with 31 routers. This table was obtained by tedious enumeration of all possible locations of 1, 2 or 3 IP routers in the tree and finding the number of occurrences that results in the same number of IP, ME, EI and M routers.

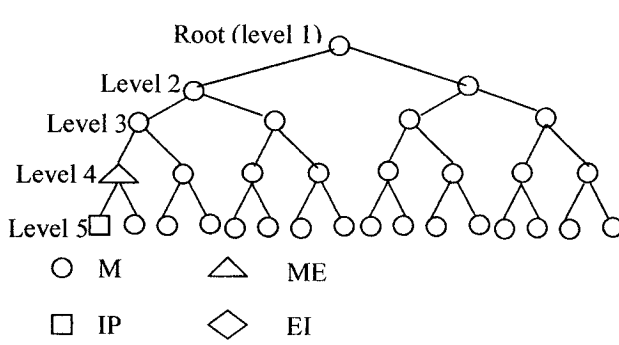


Fig. 6-1 1 IP router at level 5 in a heterogeneous MPLS network

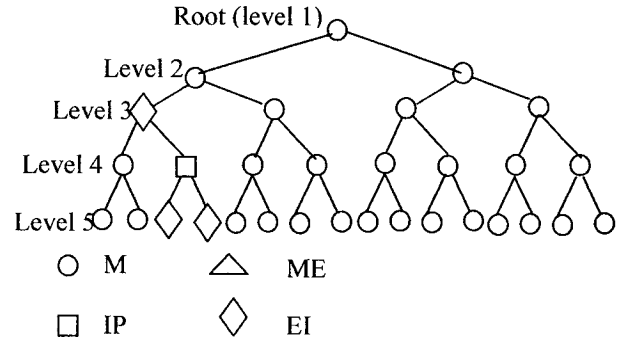


Fig. 6-2 1 IP router at level 4 in a heterogeneous MPLS network

### 6.2.2 The Analytical Model Underlying the Fair Share Policy (FSP)

Same FSP and coupled state diagrams that were used in chapter 4, will be used here. However, we have new definition to the arrival probabilities that depend on router type as follows:

For **IP routers** (type 1), the source arrival probability  $\alpha$  is actually a composite one; for instance  $\alpha_1$  can be written as:

$$\alpha_1 = \tau \alpha_1^1 + \alpha_1^2, \quad \tau = \frac{\Delta_1 + \Delta_2}{\Delta_1}$$

Where  $\Delta_1$  is the processing time at lower layers (for example MAC layer) and  $\Delta_2$  is the processing time at IP layer and  $\tau$  is the IP processing time factor (or processing factor);  $\alpha_1^1$

is the intrinsic arrival probability at the application layer (on top of IP layer),  $\alpha_1^2$  is the extra arrival probability due to IP control overhead which is used to establish the IP multicast tree. The above equation can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad (6-1)$$

where  $\xi_1$  is the IP control overhead factor (or IP factor).

Similarly for regular **MPLS routers** (type 4),  $\alpha_1$  can be written as:

$$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 \quad , \text{where } \alpha_1^1 \text{ and } \alpha_1^2 \text{ are the same as in the case of IP networks; } \alpha_1^3 \text{ is}$$

the extra arrival probability due to MPLS control overhead which is used to establish the MPLS multicast paths or tree.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2)\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad \xi_2 = \frac{\alpha_1^3}{\alpha_1^1} \quad (6-2)$$

Where  $\xi_2$  is the MPLS control overhead factor (or MPLS factor).

For **ME routers** (type 2),  $\alpha_1$  can be written as:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4$ , where  $\alpha_1^1$ ,  $\alpha_1^2$  and  $\alpha_1^3$  are the same as in the case of regular MPLS router;  $\alpha_1^4$  is the extra arrival probability due to the overhead on the MPLS router because the downstream router is an IP router.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2 + \xi_3)\alpha_1^1 \quad \xi_3 = \frac{\alpha_1^4}{\alpha_1^1} \quad (6-3)$$

Where  $\xi_3$  is ME factor.

For **EI routers** (type 3),  $\alpha_1$  can be written as:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^5$  , where  $\alpha_1^1$  ,  $\alpha_1^2$  and  $\alpha_1^3$  are the same as in the case of regular MPLS router;  $\alpha_1^5$  is the extra arrival probability due to the overhead of tunnel establishment and maintenance and also due to more packet processing is needed because of the IP routers which reside in between EI routers.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$

as:

$$\alpha_1 = (1 + \xi_1 + \xi_2 + \xi_4)\alpha_1^1 \quad \xi_4 = \frac{\alpha_1^5}{\alpha_1^1} \quad (6-4)$$

Where  $\xi_4$  is EI factor.

### 6.2.3 Reliable Multicast Cases Under Study

In this section, we will study the following four cases:

**Case 1 Without FEC or ARQ.** In the first case, we make a performance comparison between homogeneous IP networks, homogeneous MPLS networks and heterogeneous MPLS networks using FSP when DiffServ is adopted and when reliability is not considered, i.e., no FEC or ARQ would be used.  $P_{c_p}$ , which is the probability of successful delivery to next router for certain priority traffic for homogeneous networks, would be given as:

$$P_{c_p} = (1 - P_{o_p} - P_{e_p})^L, \quad p = 1, 2 \text{ or } 3, \text{ which is the priority of a certain traffic} \quad (6-5)$$

However equation 6-5 represents the probability of successful delivery to the next router in case of homogeneous networks. In case of heterogeneous networks the average probability of successful delivery to the next router will be a function of the type of router (IP, ME, EI or M) and can be found as follows:

From table 6-1, the probability of each situation (an instance of the multicast network topology, which depends on the number, and locations of IP routers) can be found using:

$$P_{s_i} = O_i * T_i^{n1} * (1 - T_i)^{N-n1} \quad (6-6)$$

Where  $O_i$  represents the number of occurrences for situation  $i$  and  $T_i$  is the probability of the number of IP routers in situation  $i$  which equals  $n1/N$ . For example the probability of

situation 9 in table 6-1 (which is the probability of having 2 IP, 0 ME, 6 EI and 23 M routers in the binary tree multicast network with 31 routers) can be expressed as:

$P_{s_9} = 64 * (2/31)^2 * (1 - (2/31))^{29}$  , with 64 be the number of occurrences of this situation due to all possible locations of the 2 IP, 0 ME, 6 EI and 23 M routers. Because of the many situations resulting from all possible selections of the IP router, it follows that if one picks a router at random from the 31 routers, the probability of that router be of type IP router is  $\theta_k = n_{i/k} / N$  where  $n_{i/k}$  is the number of routers in the situation  $i$  given they are of type  $k$  ( $k=1,2,3$  or  $4$  for IP, ME, EI and M routers respectively). Therefore, the average probability of the number of IP routers in the network can be found (from table 6-1 and equation 6-6) as:

By removing the conditioning on  $\theta_k$  by multiplying by  $P_{s_i}$  .

$$\overline{\theta_k} = \sum_{i=1}^{26} (n_{i/k} / N) * P_{s_i} \quad , k=1,2,3, \text{ and } 4 \text{ for IP, ME, EI and M routers respectively.}$$

Similarly removing the conditioning of  $P_{c_{p/k}}$  on  $\overline{\theta_k}$  obtained, the average probability of successful delivery to the next router for priority traffic  $p$  can be found as:

$$\overline{P_{c_p}} = \overline{\theta_1} * P_{c_{p/1}} + \overline{\theta_2} * P_{c_{p/2}} + \overline{\theta_3} * P_{c_{p/3}} + \overline{\theta_4} * P_{c_{p/4}} = \sum_{k=1}^4 \overline{\theta_k} * P_{c_{p/k}} \quad (6-7)$$

The total delay that a certain packet with priority  $p$  would experience from sender until it reaches the receiver is given by:

$$D_{p\_Total} = \overline{D} \overline{D_p}, \quad p = 1,2,3 \quad (6-8)$$

Where  $\overline{D}$  is the number of routers in the longest path (Depth) to the receivers (leafs of the tree), and  $\overline{D_p}$  is the average packet queuing delay (or expected number of packets) per router for certain priority traffic  $p$ . The expected number of packets for traffic with priority  $p$  can be found using:

$$\overline{E_p(n)} = \sum_{i=1}^{\max} i P_{ip} \quad , \quad p = 1,2,3 \quad (6-9)$$

Equation 6-9 actually can be evaluated using equation 4-14. The second moment of delay in units of (Packets)<sup>2</sup> can be found using:  $\overline{E_p^2(n)} = \sum_{i=1}^{\max} i^2 P_{ip}$ ,  $p = 1,2,3$

Delay jitter (standard deviation) per router for certain priority traffic  $p$  can be expressed as:  $\sigma_{xp} = \sqrt{\overline{E_p(n)^2} - (\overline{E_p(n)})^2}$ ,  $p = 1,2,3$

By assuming that total delays for all routers are statistically independent, the total delay jitter for certain priority traffic (total standard deviation) can be found using:

$$\sigma_{p\text{Total}} = D \sigma_{xp}, \quad p = 1,2,3 \quad (6-10)$$

**For homogeneous networks**, probability of multicast success (all  $N$  routers receive the multicast packet) and multicast residual packet loss probability for certain priority traffic can be found using the following equations

$$Ps_p \equiv \text{Probability of success} = Pc_p^N \quad \text{and} \quad Ploss_p = 1 - Ps_p, \quad p = 1,2,3 \quad (6-11)$$

**For heterogeneous networks**, probability of multicast success and multicast residual packet loss probability for certain priority traffic  $p$  can be found using the following equations:

$$Ps_p \equiv \text{Probability of success} = Pc_p^N \quad \text{and} \quad Ploss_p = 1 - Ps_p, \quad p = 1,2,3 \quad (6-12)$$

Where  $N$  is the total number of routers in the multicast network.

### Case 2 Using FEC only

The analysis part of this case is very similar to case 2 in subsection 4.3.2. The total delay and total delay jitter for both homogeneous and heterogeneous networks can be found using equations 6-8 and 6-10 from Case 1 of this chapter. Probability of multicast success (all  $N$  routers receive the multicast packet) and residual multicast loss probability for

certain priority traffic  $p$  for homogeneous and heterogeneous networks can be found using equations 6-11 and 6-12 respectively.

### Case 3 Using ARQ only

In case of using ARQ only,  $P_{c_p}$  for homogeneous networks and  $\overline{P_{c_p}}$  for heterogeneous networks would be used and it would be similar to case 1 of this chapter and can be found using equations 6-5 and 6-7. We have two subcases: ARQ only that uses multicast repairs and ARQ only that uses unicast repairs.

#### a) ARQ Multicast repairs

In this case upon the receipt of a NAK from one or more receivers, the sender multicast again the repair packet to all receivers. Due to the use of ARQ multicast repairs, the intrinsic arrival probability  $\alpha_p^1$  for certain priority traffic  $p$  would increase according to:

$$\alpha_p^{1'} = \alpha_p^1(1 + F_p), \quad p = 1, 2, 3 \quad (6-13)$$

$F_p$  is the number of failures for certain priority traffic  $p$ . This increase in the intrinsic arrival probability is due to that every router in the whole network receives a copy of each repair packet.

**For homogeneous networks**, the probability of success for worst case scenario for certain priority traffic  $p$  is given as:

$$P_{s_p} \equiv \text{Probability of success} = P_{c_p}^N \quad (\text{worst case scenario}) \quad (6-14)$$

The average probability of success for a certain priority  $p$  packet in a typical transmission multicast trial from sender can be calculated as:

$$P_{s_p \text{ avg}} \equiv \frac{P_{c_p}^N + P_{c_p}^{(N/2)+1} + P_{c_p}^{(N/4)+2} + P_{c_p}^{(N/8)+3} + P_{c_p}^{(N/16)+4}}{D} \quad (6-15)$$

Where  $D$  is the network depth.

Therefore, the residual loss probability (after all ARQ trials) can be found as in equation (4-32):

$$P_{\text{loss}_p} = 1 - P_{S_p}' \quad (6-16)$$

The total delay for specific priority traffic can be found as in equation (4-33):

$$D_{p\text{Total}} = (1 + F_p) \overline{D D_p} \quad (6-17)$$

Where  $F_p$  is the number of failures for priority  $p$  traffic,  $D$  is network depth and  $\overline{D_p}$  is the average packet delay per router for packet with priority  $p$ . The total delay jitter can be found using equation (6-10) from Case 1.

**For heterogeneous networks**, use  $\overline{P_{c_p}}$  formula (equation 6-7) to replace  $P_{c_p}$  in equations 6-14 and 6-15 and follow the same procedure of homogeneous networks.

#### b) ARQ Unicast repairs

Due to use of ARQ unicast repairs, the intrinsic arrival probability  $\alpha_p^1$  for certain priority traffic would change according to:

$$\alpha_p^{1'} = \alpha_p^1 \left(1 + F_p \frac{D}{N}\right) (1 + \Delta), \quad p = 1,2,3 \quad (6-18)$$

Where  $\Delta$  is the extra arrival rate due to processing of unicast repairs. This increase in the intrinsic arrival probability is due to the fact that only those routers on the path (of maximum  $D$  hops) to the router that requires repair would need the repair packet. These routers only receive a copy of repair packet so the retransmission factor would be  $(D/N)$  as compared to the multicast repairs case where we have  $(N/N)$ .

Fig. 4-22 shows the conditional probabilities tree for unicast repair for homogeneous networks. The total number of trials  $T_p$  for specific priority traffic  $p$  can be found (similar to case 3b of section 4.3.2) in a closed form for infinite number of trials as:



$$T_p = Pc_p^N + (1 - Pc_p^N) \left[ 1 + \frac{1}{Pc_p^D} \right] \quad \text{for infinite trials} \quad (6-19)$$

The number of failures  $F_p$  (or retransmissions only) for specific traffic with priority  $p$  is given as:

$$F_p = T_p - 1 = Pc_p^N + (1 - Pc_p^N) \left[ 1 + \frac{1}{Pc_p^D} \right] - 1 = \frac{1 - Pc_p^N}{Pc_p^D} \quad \text{for infinite trials} \quad (6-20)$$

The total number of trials  $T_p$  for specific priority traffic  $p$  can be found in a closed form for  $(z)$  trials as:

$$T_p = Pc_p^N + \frac{(1 - Pc_p^N)Pc_p^D}{(1 - Pc_p^D)^2} \left\{ \frac{(1 - Pc_p^D)\{1 - z(1 - Pc_p^D)^{z-1} + (z-1)(1 - Pc_p^D)^z\}}{[Pc_p^D]^2} \right\} + z(1 - Pc_p^D)^z - (1 - Pc_p^D) \quad (6-21)$$

Therefore, the number of failures (or retransmissions only) for certain priority traffic  $p$  for  $(z)$  trials can be given as:

$$F_p = T_p - 1 \quad \text{for } (z) \text{ trials} \quad (6-22)$$

Defining  $Ps_p'$  as the final probability of success for priority  $p$  traffic for infinite number of trials (which can be obtained similar to case 3b of section 4.3.2):

$$Ps_p' = 1 \quad (6-23)$$

For finite number of trials  $(z)$ ,  $Ps_p'$  can be found (similar to case 3b in section 4.3.2):

$$Ps_p' = Pc_p^N + (1 - Pc_p^N) \left[ 1 - (1 - Pc_p^D)^{z-1} \right] \quad (6-24)$$

Therefore, the residual loss probability (after all ARQ trials) is given by:

$$Ploss_p = 1 - Ps_p' \quad (6-25)$$

The total delay for specific priority traffic  $p$  is given by:

$$D_{pTotal} = (1 + F_p) \overline{D D_p} \quad (6-26)$$

The total delay jitter can be found using equation (6-10) from Case 1.

**For heterogeneous networks**, use  $\overline{Pc_p}$  formula (equation 6-7) to replace  $Pc_p$  in equations (6-19, 6-20, 6-21, 6-23, 6-24) and Fig. 4-22.

#### Case 4 Hybrid FEC/ARQ

In this case and for **homogeneous networks**,  $P_{c_p}$  would be similar to the case of FEC only, which is:

$$P_{c_p} = \sum_{i=0}^L \sum_{j=0}^{L-i} \binom{L}{i} P e_p^i \binom{L-i}{j} P o_p^j (1 - P o_p - P e_p)^{L-i-j}, p = 1,2,3 \quad (6-27)$$

Provided that  $2i + j \leq e = n - k$ ;  $r_p = \frac{k}{n}$  ;  $r_p$  is the FEC coding rate of RS code used.

The intrinsic arrival probabilities for certain traffic with priority  $p$  ( $p=1,2,3$ ) could be expressed as:

$$\begin{cases} \alpha_p' = \frac{\alpha_p^1}{r_p} (1 + F_p) & \text{(For multicast case)} \\ \alpha_p' = \frac{\alpha_p^1}{r_p} (1 + F_p \frac{D}{N}) (1 + \Delta) & \text{(For unicast case)} \end{cases} \quad (6-28)$$

Also we have 2 subcases, FEC/ARQ that uses multicast repairs and FEC/ARQ that uses unicast repairs. The analysis of hybrid FEC/ARQ would be very similar to the ARQ only case except a better value of  $P_{c_p}$  would be used as specified by equation (6-27). In addition, there would be an increase in the intrinsic arrival probabilities as in equation (6-28). Notice that the total delay jitter can be found using equation (6-10) from Case 1.

**For heterogeneous networks** in the hybrid FEC/ARQ case , we use  $\overline{P_{c_p}}$  formula (equation 6-7) to replace  $P_{c_p}$  in all equations of the ARQ case.

Six different programs were developed for the purpose of calculations and to solve the involved and non-linear set of equations in order to find the performance measures (total delay, delay jitter and residual loss probability). These programs are kept running until the set of equations converge. After convergence, performance measures like total packet delay, total packet delay jitter and the residual packet loss probability are calculated. See Flowchart of Fig. 6-3.

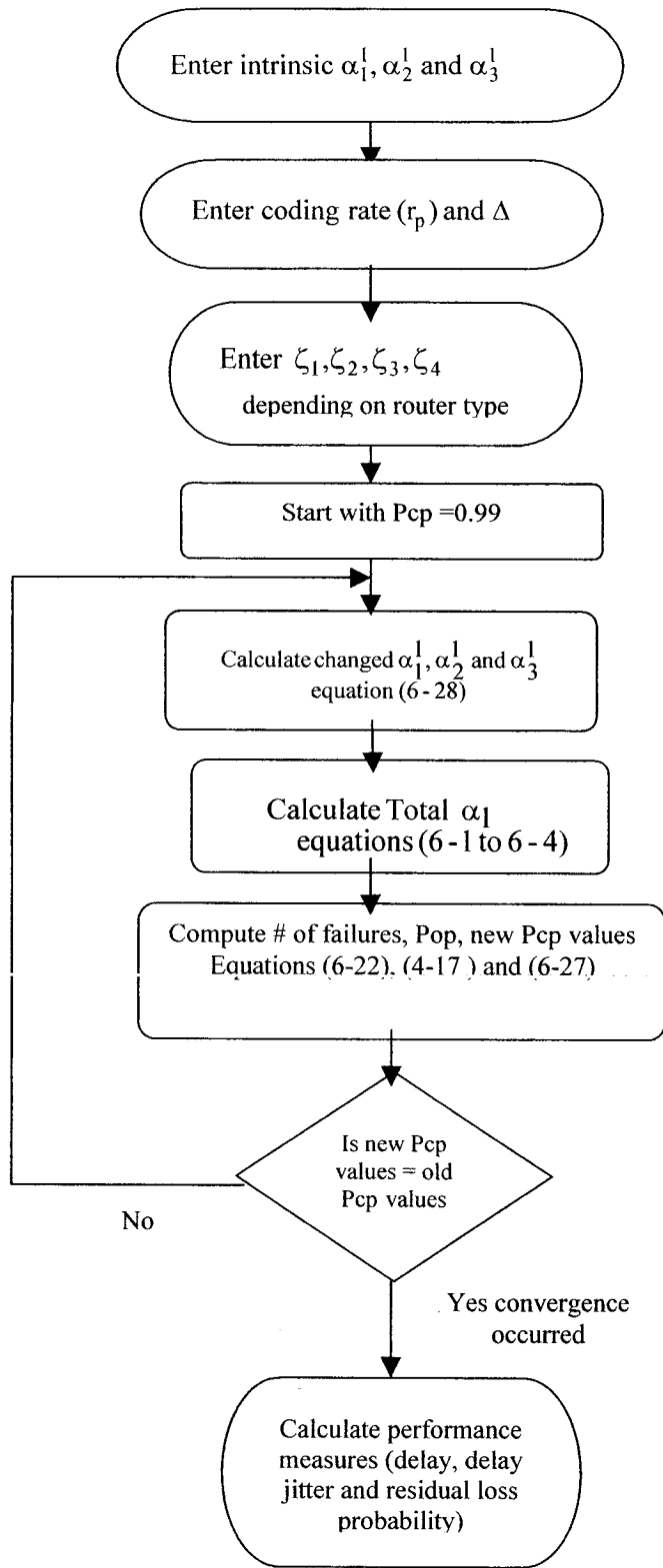


Fig. 6-3 Flowchart of program which calculate performance measures of Hybrid FEC/ARQ unicast repairs

## 6.2.4 Analysis Results

Figs. 6-4 to 6-6 show the performance comparisons between homogeneous IP network (with each router has IP1, IP2 and IP3 sources), homogeneous MPLS network (with each router has M1, M2 and M3 sources) and heterogeneous MPLS network (with each router has H1, H2 and H3 sources). In these figures no FEC or ARQ is applied. Figs. 6-4 to 6-6 show the total packet delay, total delay jitter, and the residual loss probability for all sources versus IP factor for small processing factor ( $\tau$ ). Figs. 6-4, 6-5 and 6-6 show that MPLS sources will have the best performance in terms of total packet delay, total delay jitter and residual loss probability; on the other hand the heterogeneous sources will have the worst performance.

Figs. 6-7, 6-8 and 6-9 consider the performance comparisons between the three types of multicast networks when FEC mechanism only is applied. The tendencies of Figs. 6-7 to 6-9 are similar to the case of Figs. 6-4 to 6-6. However, when using FEC there would be a slight increase in the total packet delay for all sources compared to without using FEC or ARQ due to the increase in the intrinsic arrival probabilities because of the FEC operation. However, the residual packet loss probability for all sources would decrease due to the use of improved  $P_c$  value in the case of FEC only.

Figs. 6-10 to 6-12 consider the performance comparisons between the three types of multicast networks when ARQ only with multicast repairs mechanism is applied. The tendencies of Figs. 6-10 to 6-12 are very similar to the case of Figs. 6-4 to 6-6.

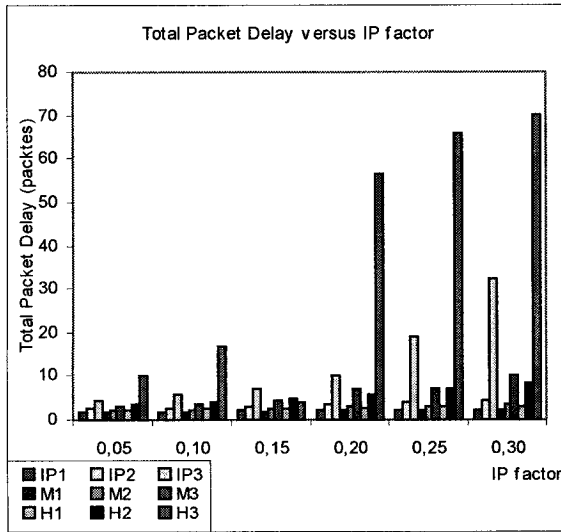
Figs. 6-13 to 6-15 consider the performance comparisons between the three types of multicast networks when ARQ only with unicast repairs mechanism is applied. The tendencies of Figs. 6-13 to 6-15 are very similar to the case of Figs. 6-10 to 6-12. Using

ARQ only would have the worst total packet delay for all IP, MPLS and heterogeneous sources compared to without FEC or ARQ (case 1) or FEC only (case 2) due to retransmission requests. However, using ARQ only would improve the residual packet loss probability in a noticeable manner. ARQ with unicast repairs would be better than ARQ with multicast repairs in terms of residual packet loss probability but worst than it in terms of total packet delay.

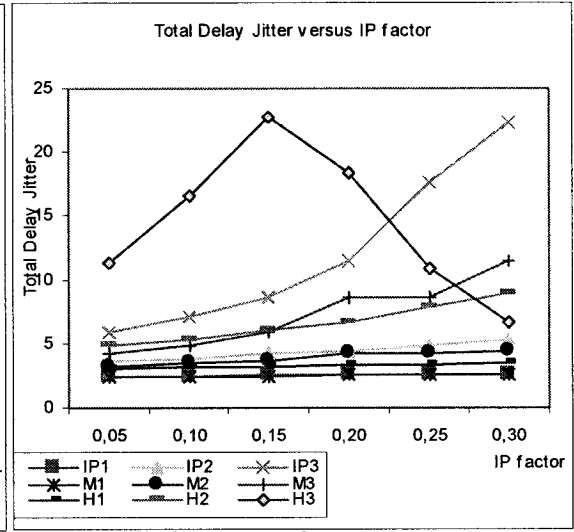
In Figs. 6-4 to 6-15 MPLS factor was constant and relatively small; explaining why MPLS performance was better or very similar to IP performance. However, in the following figures we will study the effects of MPLS factor on MPLS performance. Figs. 6-16 to 6-18 show the performance comparisons between the three types of multicast networks when hybrid FEC/ARQ with multicast repairs mechanism is applied. Fig. 6-16 to 6-18 show the total packet delay, total delay jitter, and the residual packet loss probability for all sources versus MPLS factor. These figures show that IP sources will have the best performance in terms of total packet delay, total delay jitter and residual packet loss probability; on the other hand the heterogeneous sources will have the worst performance. Figs. 6-19 to 6-21 show the performance comparisons between the three types of multicast networks when hybrid FEC/ARQ with unicast repairs mechanism is applied. The tendencies of Figs. 6-19 to 6-21 are very similar to the case of Figs. 6-16 to 6-18.

Using hybrid FEC/ARQ would have the best performance in terms of residual packet loss probability among all schemes and the worst performance in terms of total packet delay among all schemes. In addition to that the hybrid FEC/ARQ with unicast repairs

performs better than hybrid FEC/ARQ with multicast repairs in terms of residual packet loss probability but worse than it in terms of total packet delay.



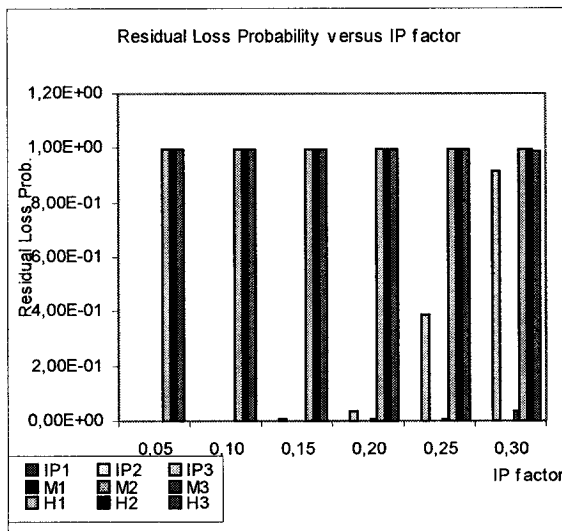
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500$



$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500$

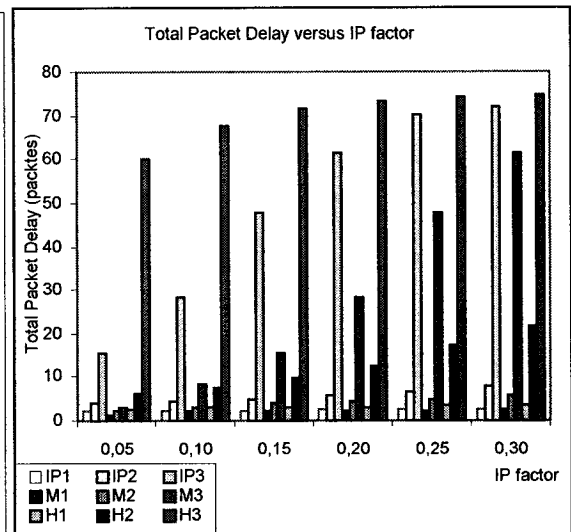
Fig. 6-4 Total packet delay versus IP factor (No FEC or ARQ and small  $\tau$ )

Fig. 6-5 Total delay jitter versus IP factor (No FEC or ARQ and small  $\tau$ )



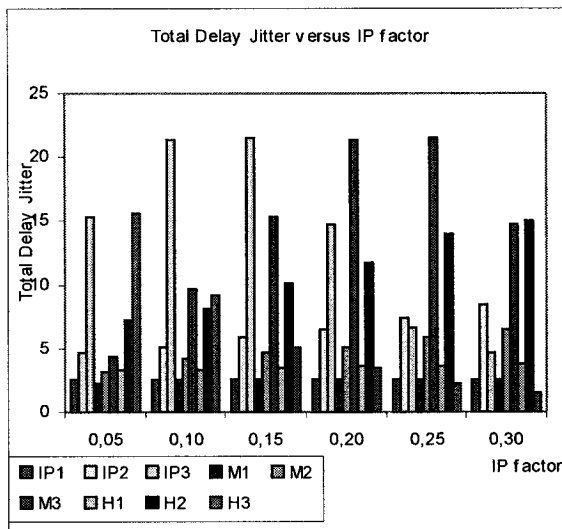
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500$

Fig. 6-6 Residual loss probability versus IP factor (No FEC or ARQ and small  $\tau$ )



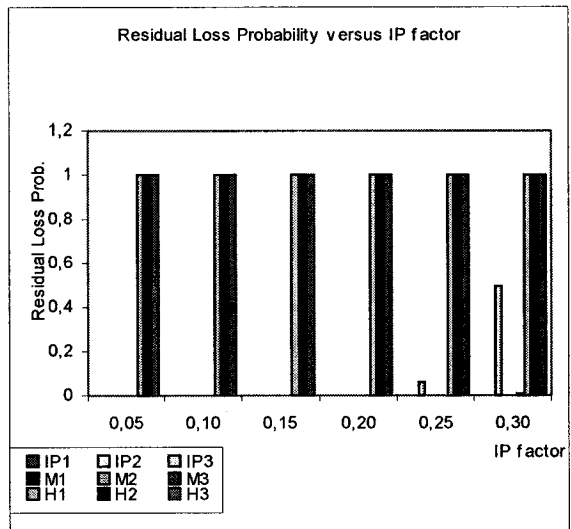
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, r = 223/255$

Fig. 6-7 Total packet delay versus IP factor (FEC only and small  $\tau$ )



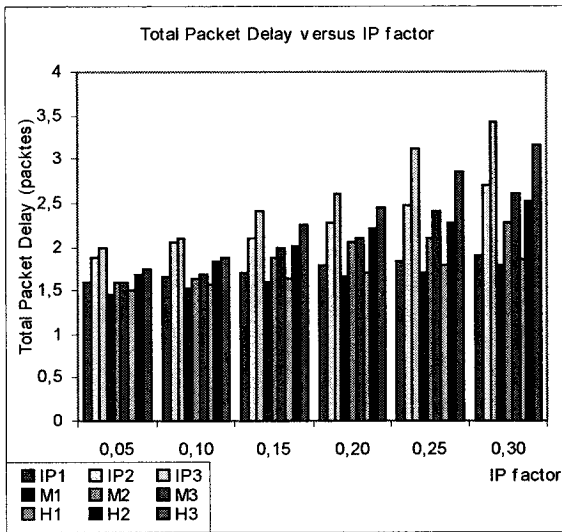
$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, r = 223/255$

Fig. 6-8 Total delay jitter versus IP factor (FEC only and small  $\tau$ )



$\alpha_1^1 = 0.3, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15,$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, r = 223/255$

Fig. 6-9 Residual loss probability versus IP factor (FEC only and small  $\tau$ )

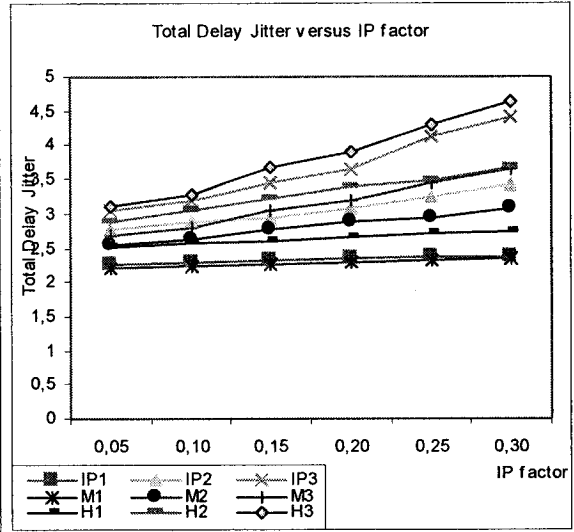


$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1,$$

$$D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$$

$$\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$$

Fig. 6-10 Total packet delay versus IP factor (ARQ multicast and small  $\tau$ )

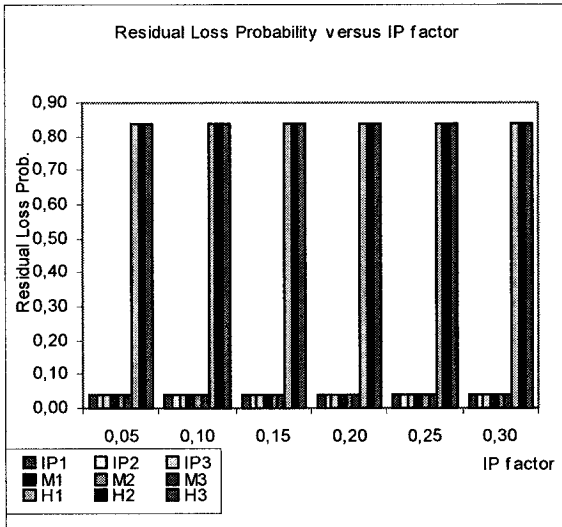


$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1,$$

$$D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$$

$$\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$$

Fig. 6-11 Total delay jitter versus IP factor (ARQ multicast and small  $\tau$ )

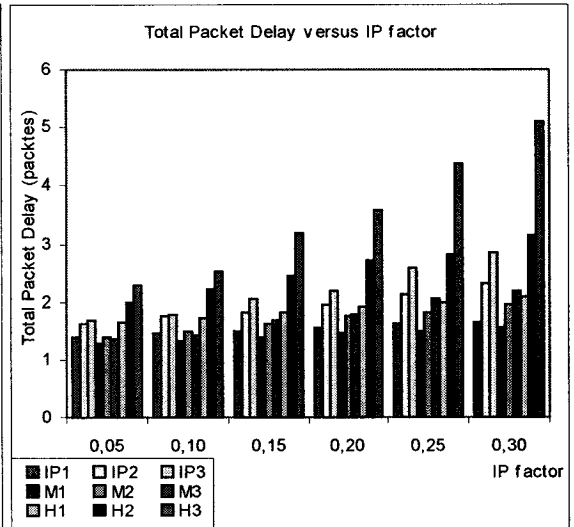


$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1,$$

$$D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$$

$$\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$$

Fig. 6-12 Residual loss probability versus IP factor (ARQ multicast and small  $\tau$ )



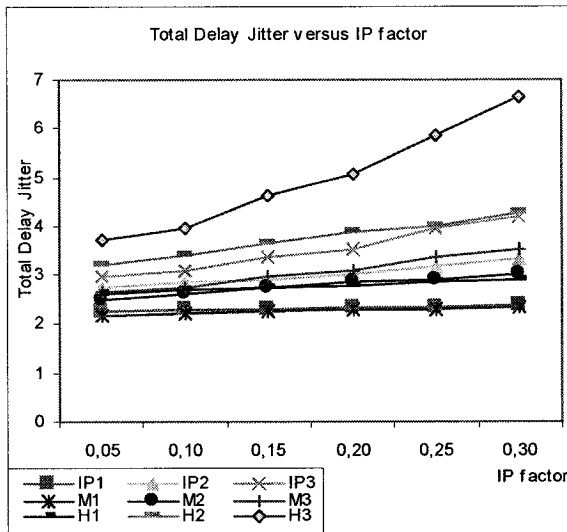
$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \Delta = 0.1$$

$$D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$$

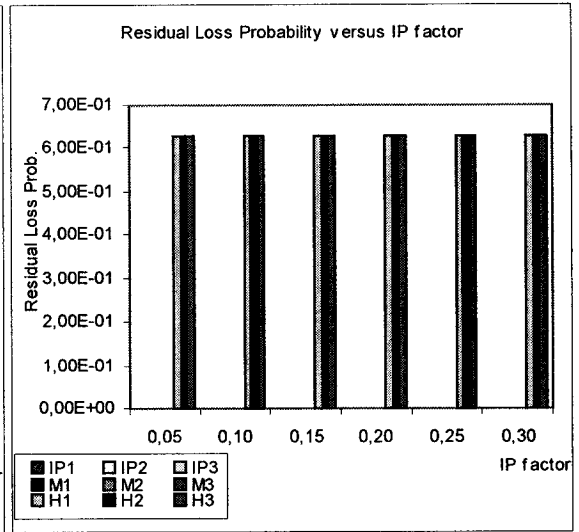
$$\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$$

Fig. 6-13 Total packet delay versus IP factor (ARQ unicast and small  $\tau$ )





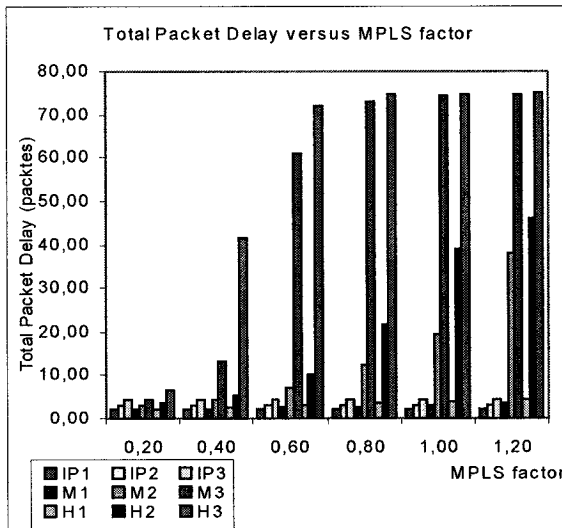
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \Delta = 0.1$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, \Delta = 0.1$   
 $D = 5, B = 30, \xi_2 = 0.1, \xi_3 = 0.05$   
 $\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$

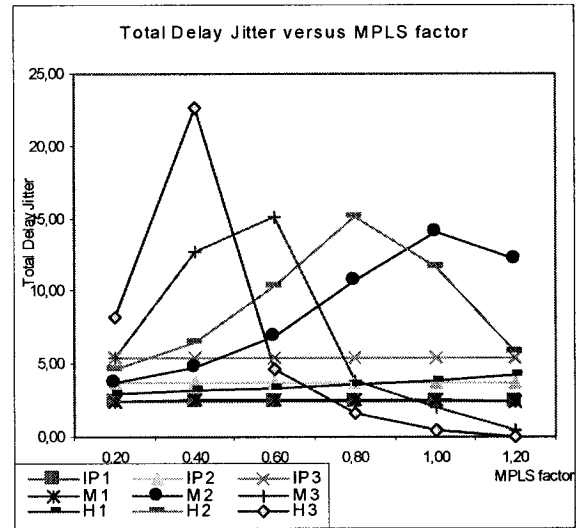
Fig. 6-14 Total delay jitter versus IP factor (ARQ unicast and small  $\tau$ )

Fig. 6-15 Residual loss probability versus IP factor (ARQ unicast and small  $\tau$ )



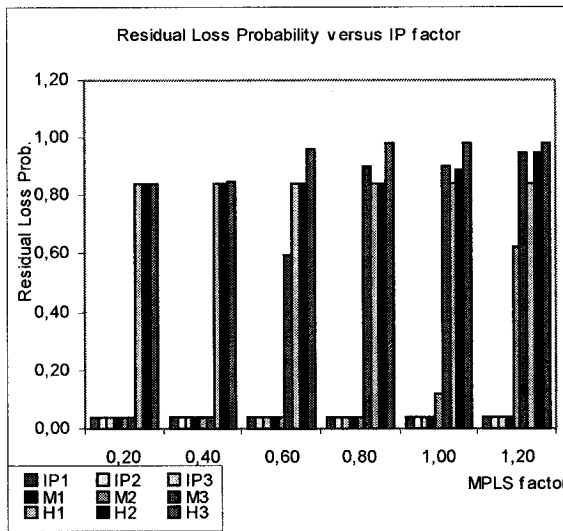
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05$   
 $\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$

Fig. 6-16 Total packet delay versus MPLS factor (hybrid multicast)



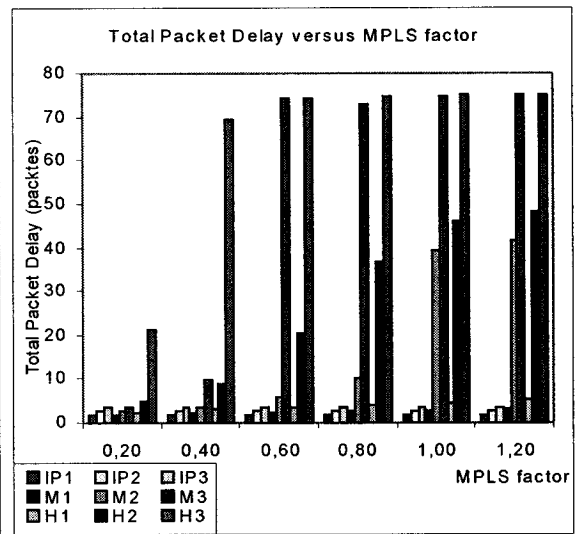
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05$   
 $\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$

Fig. 6-17 Total delay jitter versus MPLS factor (hybrid multicast)



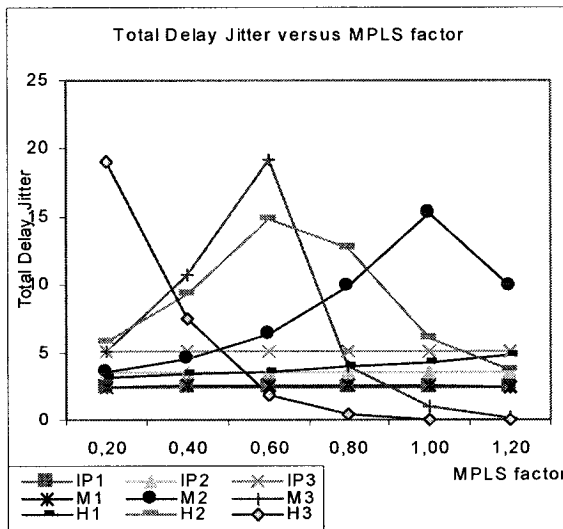
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, z = 2$

Fig. 6-18 Residual loss probability versus MPLS factor (hybrid multicast)



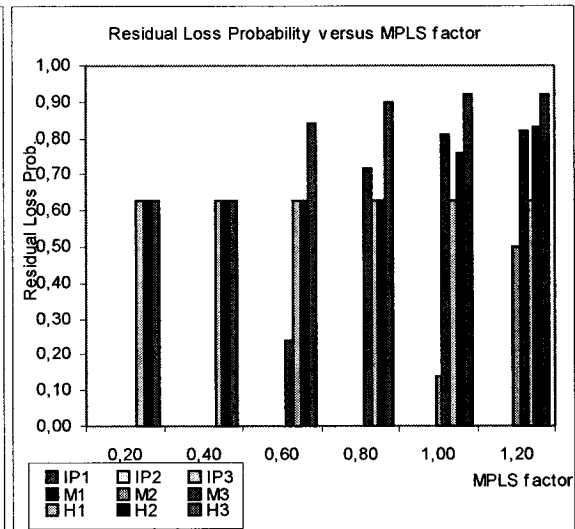
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05, \Delta = 0.1$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, z = 2$

Fig. 6-19 Total packet delay versus MPLS factor (hybrid unicast)



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05, \Delta = 0.1$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, z = 2$

Fig. 6-20 Total delay jitter versus MPLS factor (hybrid unicast)



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.2, \xi_3 = 0.05, \Delta = 0.1$   
 $\xi_4 = 0.1 \tau = 1.2, L = 500, z = 2$

Fig. 6-21 Residual loss probability versus MPLS factor (hybrid unicast)

Table 6-1 The distribution table of the existence possibilities of 1 IP, 2 IP or 3 IP routers in a heterogeneous MPLS network with 31 routers.

Situation # (Si)	Combinations (Ci)	One IP Router (occurrences) Oi	Two IP Routers (occurrences) Oi	Three IP Routers (occurrences) Oi
1	1 IP, 1 ME, 0 EI, 29 M	16		
2	1 IP, 0 ME, 3 EI, 27 M	14		
3	2 IP, 1 ME, 0 EI, 28 M		8	
4	2 IP, 2 ME, 0 EI, 27 M		112	
5	2 IP, 0 ME, 2 EI, 27 M		16	
6	2 IP, 0 ME, 3 EI, 26 M		16	
7	2 IP, 0 ME, 4 EI, 25 M		12	
8	2 IP, 0 ME, 5 EI, 24 M		15	
9	2 IP, 0 ME, 6 EI, 23 M		64	
10	2 IP, 1 ME, 3 EI, 25 M		192	
11	3 IP, 1 ME, 0 EI, 27 M			8
12	3 IP, 2 ME, 0 EI, 26 M			112
13	3 IP, 3 ME, 0 EI, 25 M			448
14	3 IP, 0 ME, 8 EI, 20 M			70
15	3 IP, 0 ME, 9 EI, 19 M			120
16	3 IP, 1 ME, 3 EI, 24 M			96
17	3 IP, 1 ME, 2 EI, 25 M			224
18	3 IP, 2 ME, 3 EI, 23 M			1120
19	3 IP, 0 ME, 3 EI, 25 M			232
20	3 IP, 0 ME, 4 EI, 24 M			112
21	3 IP, 1 ME, 5 EI, 22 M			208
22	3 IP, 0 ME, 5 EI, 23 M			158
23	3 IP, 1 ME, 6 EI, 21 M			800
24	3 IP, 0 ME, 7 EI, 21 M			132
25	3 IP, 0 ME, 6 EI, 22 M			156
26	3 IP, 1 ME, 4 EI, 23 M			64
<b>Total number of occurrences</b>		30	435	4060

### **6.3 Reliable QoS Multicast for DiffServ Over Hybrid FEC/ARQ IP and MPLS Heterogeneous Networks**

The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). In this chapter, we defined **heterogeneity** as “ **the coexistence of different types of routers with different capabilities in the same network**”.

In this section, we will have four types of networks and these are: homogeneous MPLS multicast network, homogeneous IP multicast network, heterogeneous MPLS multicast network and heterogeneous IP multicast network, each of them with the possibility of existence of more than one Domain Router (DR) router at different locations in the multicast tree.

In addition to that, in this thesis we compare the QoS performance of the four types of multicast networks mentioned above when reliable FEC/ARQ with unicast repair multicasting is used, given their particular constraints. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. We present a fair share policy (FSP) and by taking the above constraints into consideration, we evaluate the QoS performance for a typical binary tree in the various cases mentioned. We also consider Differentiated Services; i.e. traffics with different priority classes when reliable multicast is used. Analysis tools will be used to evaluate our fair share policy (FSP).

### **6.3.1 Heterogeneous MPLS and Heterogeneous IP based networks**

In addition to the homogeneous MPLS and IP multicast networks, heterogeneous MPLS and IP multicast networks will be studied. Fig. 5-1 shows an example of a complete homogeneous binary multicast tree with the root, which is the nearest router to the sender or the Rendezvous point; the leafs, which are the routers with receivers underneath them. As shown in the figure the depth of this tree is 5 and the total number of routers is 31. This tree could represent either an MPLS multicast tree or an IP multicast tree. Moreover, the results of this section apply to trees of larger sizes.

In the heterogeneous MPLS multicast network, the network is assumed to be an MPLS network with the existence of some MPLS routers with Domain Router (DR) capability. Similarly in the heterogeneous IP multicast network, the network is assumed to be an IP network with the existence of some IP routers with Domain Router (DR) capability. The rest of the routers in the heterogeneous MPLS or heterogeneous IP multicast networks do not have the DR capability, which means that they cannot correct errors or retransmit the repair packet themselves. However, in case of error or loss, they can request a repair packet from the DR of the subtree they belong too. The existence of the DR creates a subtree underneath this DR, and in case of error or loss, all the routers that belong to this subtree request the repair packet from their corresponding DR rather than the root (or the sender) of the multicast session. If the current DR does not have the repair packet either because it has a limited cache of memory or because the repair packet is too old, it will ask its parent DR for this repair packet and so on until we reach the root

(or the sender). The root (or the sender) is assumed to be a DR and always have the repair packet.

Fig. 6-22 shows one possibility (occurrence) when 2 DR routers exist in a heterogeneous MPLS or heterogeneous IP multicast network. One DR at the level 1 (the root) and the other at level 3 in a binary multicast network with depth = 5. As one can see there are two subnets in this network. One is subnet 1 under DR 1, which has a depth of 5, and there are 25 routers that belong to subnet 1 including DR 2. The other subnet is subnet 2 under DR 2, which has a depth of 3 and there are 6 routers that belong to subnet 2. There are 4 occurrences of this possibility as clarified in table 6-2 (entry 3), and due to space limitations we don't show them all.

Fig. 6-23 shows another possibility (occurrence) when 3 DR routers exist in a heterogeneous MPLS or heterogeneous IP multicast network. One DR at the level 1 (the root), one DR at level 2 and the last DR at level 3 in a binary multicast network with depth = 5. As one can see there are three subnets in this network. One is subnet 1 under DR 1, which has a depth of 5, and there are 17 routers that belong to subnet 1 including DR 2. The other subnet is subnet 2 under DR 2, which has a depth of 4 and there are 8 routers that belong to subnet 2 including DR 3. Finally subnet 3 under DR 3, which has a depth of 3 and there are 6 routers that belong to subnet 3. There are 4 occurrences of this possibility as clarified in table 6-2 (entry 19), and due to space limitations we don't show them all.

Table 6-2 shows the distributions of the existence possibilities of 1 DR, 2 DR routers in addition to the DR router at the root in a heterogeneous MPLS or IP multicast network with 31 routers (depth =5).

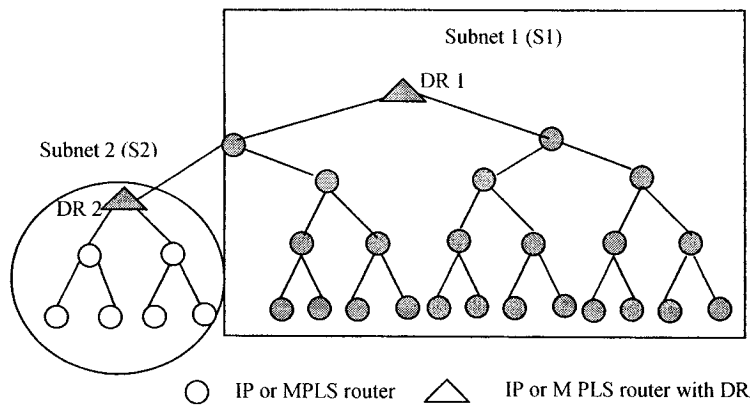


Fig. 6-22 Heterogeneous binary multicast tree with 2 DRs and 2 Subnets

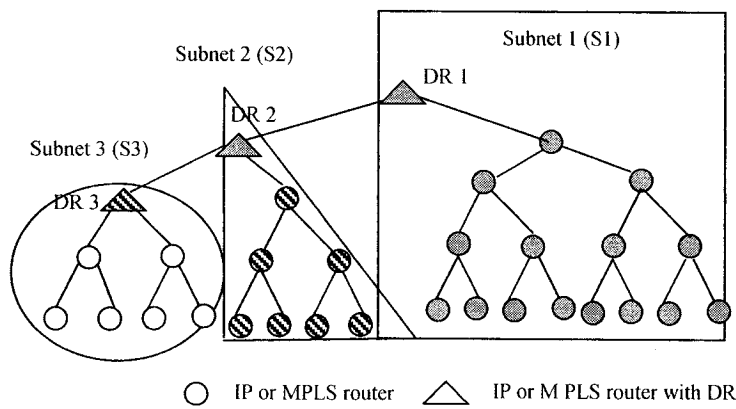


Fig. 6-23 Heterogeneous binary multicast tree with 3 DRs and 3 Subnets

### 6.3.2 The Analytical Model Underlying the Fair Share Policy (FSP)

The same analytical model shown in Fig. 4-1 and the same discrete coupled state diagrams shown in Fig. 4-2 will be used through out the analysis part of this section.

Table 6-2 The distribution table of the existence possibilities of 1 DR, 2 DR in addition to DR at root in a heterogeneous MPLS or IP network with 31 nodes.

Situation # (Sj)	Combinations (Cj)	One DR Router (occurrences) Oj	Two DR Routers (occurrences) Oj
1	31 S1, 0 S2	16	
2	29 S1, 2 S2	8	
3	25 S1, 6 S2	4	
4	17 S1, 14 S2	2	
5	31 S1, 0 S2, 0 S3		120
6	27 S1, 2 S2, 2 S3		28
7	19 S1, 6 S2, 6 S3		6
8	3 S1, 14 S2, 14 S3		1
9	29 S1, 1 S2, 0 S3		16
10	29 S1, 2 S2, 0 S3		112
11	25 S1, 6 S2, 0 S3		16
12	25 S1, 6 S2, 0 S3		48
13	17 S1, 14 S2, 0 S3		16
14	17 S1, 14 S2, 0 S3		16
15	25 S1, 4 S2, 2 S3		8
16	23 S1, 6 S2, 2 S3		24
17	17 S1, 12 S2, 2 S3		8
18	15 S1, 14 S2, 2 S3		8
19	17 S1, 8 S2, 6 S3		4
20	11 S1, 14 S2, 6 S3		4
<b>Total number of occurrences</b>		30	435

### 6.3.3 Reliable FEC/ARQ Multicast Case using Unicast Repair Packets

A hybrid FEC/ARQ strategy should be used where a combination of FEC for the most frequent error patterns, together with error detection and retransmission for the less likely error patterns is more efficient than ARQ alone. In this case when FEC fails to correct errors at the receiver the receiver sends a NAK to the sender to retransmit the data in error. This hybrid FEC/ARQ strategy clearly carries the potential for improving throughput in two-way systems subject to a high channel error rate.



Using FEC, there are  $\binom{L}{i}$  ways to have errors in  $i$  bytes out of  $L$  bytes in the multicast packet received at a certain router. However, once we have  $i$  bytes in errors, the number of ways of selecting the location of lost bytes would be  $\binom{L-i}{j}$ .  $P_{c_p}$  which is the probability of successful delivery to next router for certain priority traffic  $p$ , is then given by:

$$P_{c_p} = \sum_{i=0}^L \sum_{j=0}^{L-i} \binom{L}{i} P_{e_p}^i \binom{L-i}{j} P_{o_p}^j (1 - P_{o_p} - P_{e_p})^{L-i-j}, p = 1,2,3 \quad (6-29)$$

Provided that  $2i + j \leq e = n - k$ ;  $r_p = \frac{k}{n}$ ; where  $k$  is the number of original data symbols,  $n$  is the total number of symbols after applying FEC encoding,  $r_p$  is the FEC coding rate and  $e$  is the total number of errors and losses (called erasures in FEC terminology) which defines the capability of the code used.  $P_{o_p}$  is the byte overflow for a certain priority traffic  $p$  and  $P_{e_p}$  is the byte error probability for a certain priority traffic  $p$ .

With unicast repair mechanism, if the sender receives a NAK from one or more receivers, it resends the repair packet to only the receivers who did not receive the packet correctly in a unicast manner. The multicast repairs method is simpler than the unicast repairs method and requires less overhead; however the multicast repairs method consumes much more bandwidth. In this section we will evaluate the reliable multicasting performance using the unicast repair method only. Due to use of FEC/ARQ unicast repair mechanism, the intrinsic arrival probability for certain priority traffic  $\alpha_p^1$  would change according to:

$$\alpha_p^{1'} = \frac{\alpha_p^1}{r_p} \left(1 + F_p \frac{D}{N}\right) (1 + \Delta) \quad (6-30)$$

Where  $\Delta$  is the extra arrival rate due to processing of unicast repairs. This increase in the intrinsic arrival probability is due to the fact that only those routers on the path (of maximum  $D$  hops) to the router that requires repair would need the repair packet. These

routers only receive a copy of repair packet so the retransmission factor would be  $(D/N)$  as compared to the multicast repair case where we have  $(N/N)$ .  $r_p$  is the FEC coding rate for priority  $p$  traffic.

Following the same procedure in section 4.3.2 (cases 3b and 4b), one can obtain the residual multicast loss and the total packet delay as:

The residual multicast loss probability for specific priority traffic  $p$  is given by:

$$P_{\text{loss } p} = 1 - P_{s_p} \quad (6-31)$$

The total delay for specific priority traffic  $p$  is given by:

$$D_{p\text{Total}} = (1 + F_p) D \overline{D}_p \quad (6-32)$$

Where  $F_p$  is the average number of failures for priority traffic  $p$ ,  $D$  is network depth and  $\overline{D}_p$  is the average packet delay per router for traffic with priority  $p$  (in terms of packets), which can be obtained from equation (4-14).

**For homogeneous networks**, equations (6-31) and (6-32) are used to calculate the residual packet loss probability and the total packet delay respectively.

**For heterogeneous networks**, the average packet delay of sub-tree  $i$  for priority traffic  $p$  can be found as:

$$\overline{D}_{i,p} = (1 + F_{i,p}) D_{i,p} * D_i + \Delta D_{i,p} \quad (6-33)$$

Where  $F_{i,p}$  is the average number of failures in sub-tree  $i$  for priority traffic  $p$ ,  $D_{i,p}$  is the packet delay per router in sub-tree  $i$  for priority traffic  $p$  (in terms of packets), which can be obtained from equation (4-14), and  $D_i$  is the depth of sub-tree  $i$ . Notice that the term  $\Delta D_{i,p}$  represents the extra delay the packet endures from the providing DR to the current router requesting the repair packet, which can be given through the following recursive equation as:

$$\overline{\Delta D_{i,p}} = \sum ((1 - Ph_i) dis_i Ph_{i-1}) * D_{i,p} \quad (6-34)$$

$Ph_i$  is the probability that the current DR router has the repair packet. If the current DR doesn't have the packet with probability  $(1 - Ph_i)$  either because of buffer overflow or when the requested repair packet is too old; then the current DR will request this repair packet from its parent DR on the multicast path provided this parent DR has the repair packet with probability  $Ph_{i-1}$  multiplied by the distance (in terms of number of links) between the current DR and the parent DR  $dis_i$  and so on in a gratuitous mode until we reach the root (or the sender).  $D_{i,p}$  is the packet delay per router in sub-tree  $i$  for priority traffic  $p$  (in terms of packets), which can be obtained from equation (4-14).

$Ph_i$  can be written using the following recursive equation:

$$Ph_i = (1 - Pov_i) + Pov_i Ph_{i-1} (Pc_i)^{dis_i} \quad (6-35)$$

In equation (6-35), the probability of having the repair packet in the current DR ( $i$ ) router  $Ph_i$  would be  $(1 - Pov_i)$  provided that this DR router does not have buffer overflow. Otherwise, if the current DR router overflow with probability  $Pov_i$ , the current DR router ( $i$ ) will request the repair packet from its parent DR router provided that this parent DR ( $i-1$ ) have the packet  $Ph_{i-1}$  and it delivers this repair packet to the requesting DR router successfully through the whole distance between them  $(Pc_i)^{dis_i}$ . Notice that we assume  $Ph_1 = 1$  which means that the root (or the sender) DR always have the repair packets.

Because of the many situations resulting from all possible types of the DR routers, it follows that if one picks a router at random from the 31 routers, the probability of that router being under DR router ( $i$ ) is:

$$\theta_{i,j} = n_{i,j} / N \quad (6-36)$$

$i$  is the sub-tree index and  $i=1,2,3$  and  $j$  is the situation number index;  $n_{i,j}$  is the number of routers in sub-tree  $i$  and situation  $j$ .  $N$  is the total number of routers in the multicast tree.

The final total packet delay in a multicast network for certain priority traffic  $p$ , can be given as:

$$D_{pTotal} = \left( \sum_{j=1}^S \overline{D_{p,j}} * O(j) \right) / \sum_{j=1}^S O(j) \quad (6-37)$$

Where  $j$  indicates the situation number in table 6-2,  $O(j)$  is the number of occurrences for situation  $j$  which also can be found using table 6-2.  $S$  is the total number of situations and  $\overline{D_{p,j}}$  is situation  $j$  average packet delay for traffic priority  $p$  which can be given as:

$$\overline{D_{p,j}} = \sum_{i=1}^3 \theta_{i,j} * \overline{D_{i,p,j}} + \theta_{3,j} * \overline{D_{3,p,j}} \quad (6-38)$$

Where  $\theta_{1,j}$ ,  $\theta_{2,j}$  and  $\theta_{3,j}$  can be obtained using equation (6-36).  $\overline{D_{1,p,j}}$ ,  $\overline{D_{2,p,j}}$  and  $\overline{D_{3,p,j}}$  are the average packet delays for sub-trees ( $i=1,2,3$ ) respectively and for priority traffic  $p$  in situation  $j$  which can be calculated using equation (6-33)

Similarly, we can find the residual loss probability in heterogeneous networks. The residual packet probability of success in a multicast network for certain priority traffic  $p$ , can be given as:

$$\overline{P_{s_p}} = \left( \sum_{j=1}^S \overline{P_{s_p,j}} * O(j) \right) / \sum_{j=1}^S O(j) \quad (6-39)$$

Where  $j$  indicates the situation number in table 6-2,  $O(j)$  is the number of occurrences for situation  $j$  which also can be found using table 6-2,  $S$  is the total number of situations and

$\overline{Ps_{p,j}}$  is situation j average probability of success for traffic priority p which can be written as:

$$\overline{Ps_{p,j}} = \sum_{j=1}^S \theta_{1,j} * \overline{Ps_{1,p,j}} + \theta_{2,j} * \overline{Ps_{2,p,j}} + \theta_{3,j} * \overline{Ps_{3,p,j}} \quad (6-40)$$

$\overline{Ps_{i,p,j}}$  is the average probability of success in the multicast sub-tree i for priority traffic p in the situation number j of table 6-2 which can be given as:

$$\overline{Ps_{i,p,j}} = \overline{Ps_{i,p,j}} Ph_i + (1 - Ph_i) Ph_{i-1} (Pc_i)^{dis_i} \quad (6-41)$$

$\overline{Ps_{i,p,j}}$  is the probability of success in the multicast sub-tree i for priority traffic p in the situation number j of table 6-2 which can be obtained using equation (4-40).

The residual packet loss probability for priority traffic p in heterogeneous multicast networks can be calculated using equation (6-39) as:

$$Ploss_p = 1 - \overline{Ps_p} \quad (6-42)$$

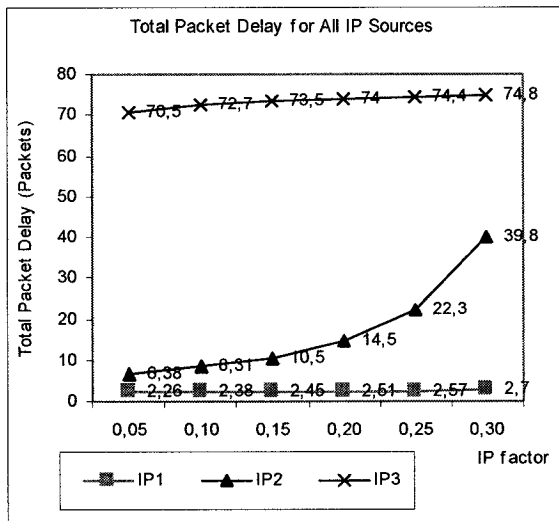
### 6.3.4 Analysis Results

Figs. 6-24 and 6-25 show the performance comparisons between IP sources (IP1, IP2 and IP3 of each router) and MPLS sources (M1, M2 and M3 of each router) respectively in homogeneous multicast trees when a hybrid FEC/ARQ with unicast repair mechanism is applied. Fig. 6-24 shows the total packet delay for all IP sources versus IP factor and for large processing factor ( $\tau$ ). Fig. 6-25 shows the total packet delay for all MPLS sources versus IP factor. As shown in Figs. 6-24 and 6-25 when the processing factor ( $\tau$ ) increases MPLS will have superiority over IP in terms of the total packet delay especially for low priority traffics. Similarly and as shown in Figs. 6-26 and 6-27 when the

processing factor ( $\tau$ ) increases MPLS will have superiority over IP in terms of the residual packet loss probability especially for low priority traffics.

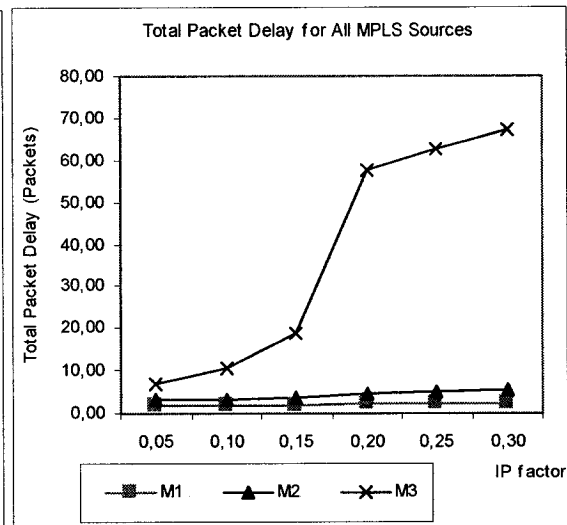
Figs. 6-28 to 6-31 show similar comparisons and have similar tendencies to Figs. 6-24 to 6-27, but for heterogeneous IP and MPLS multicast trees when hybrid FEC/ARQ with unicast repair mechanism is applied. However, a comparison between Figs. 6-28 to 6-31 and 6-24 to 6-27 shows that the use of multiple DRs in multicast network will enhance the network performance in terms of total packet delay and residual packet loss probability for all IP and MPLS sources.

In Figs. 6-24 to 6-31 MPLS factor was constant and relatively small; explaining why MPLS performance was better than IP performance. However, in the following figures we will study the effects of MPLS factor on MPLS performance. Figs. 6-32 and 6-33 show the total packet delay performance comparisons between IP sources and MPLS sources respectively in heterogeneous multicast trees. Figs. 6-34 and 6-35 show the residual packet loss probability comparisons between IP sources and MPLS sources respectively in heterogeneous multicast trees. As shown in these figures, the total packet delay and the residual packet loss probability in the case of IP (which is constant) is less than MPLS for all sources. This means when the extra arrival rate due to MPLS control overhead used to establish MPLS multicast paths (or tree) increases, IP will perform better than MPLS especially when the intrinsic arrival traffics increase.



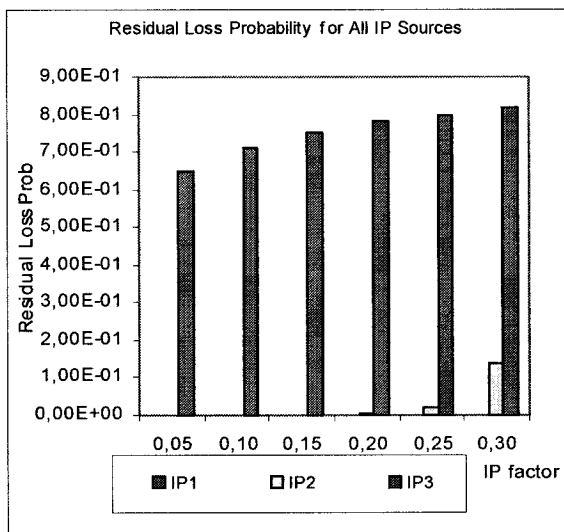
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-24 Total packet delay for IP sources versus IP factor (homogeneous networks)



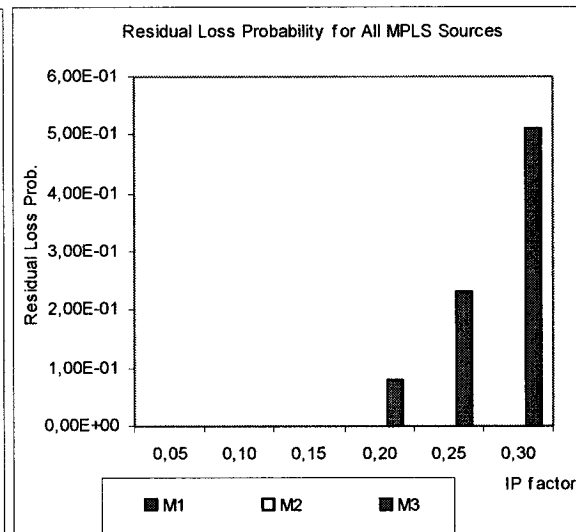
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-25 Total packet delay for MPLS sources versus IP factor (homogeneous networks)



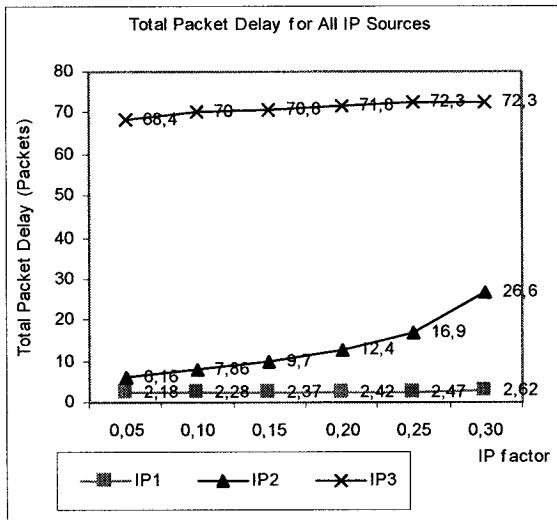
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-26 Residual loss probability for IP sources versus IP factor (homogeneous networks)



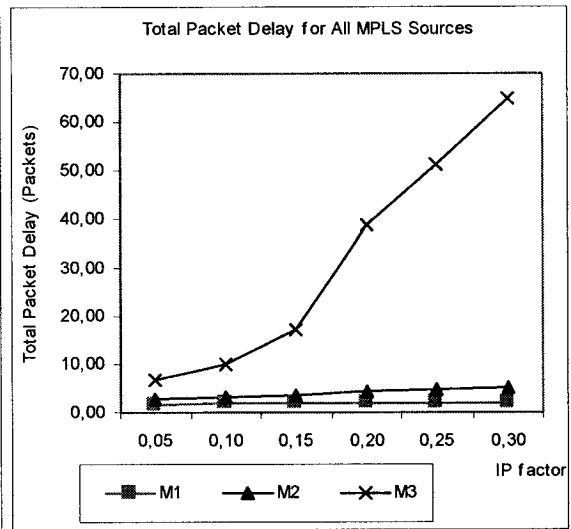
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-27 Residual loss probability for MPLS sources versus IP factor (homogeneous networks)



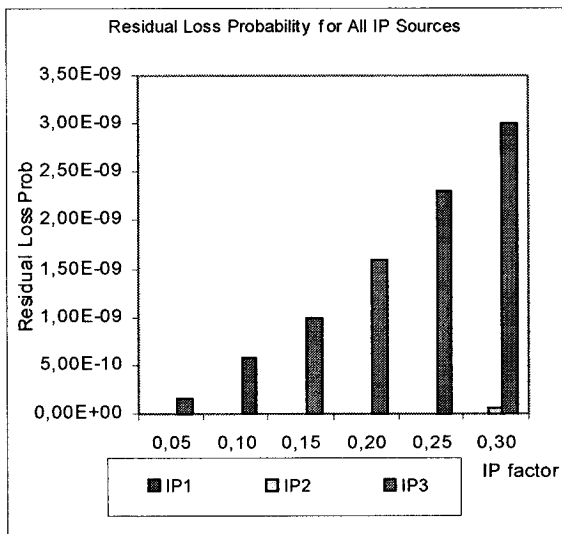
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-28 Total packet delay for IP sources versus IP factor (heterogeneous networks)



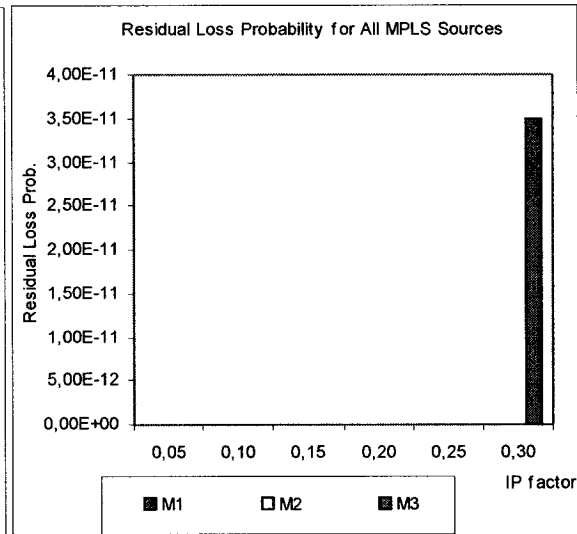
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

Fig. 6-29 Total packet delay for MPLS sources versus IP factor (heterogeneous networks)



$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

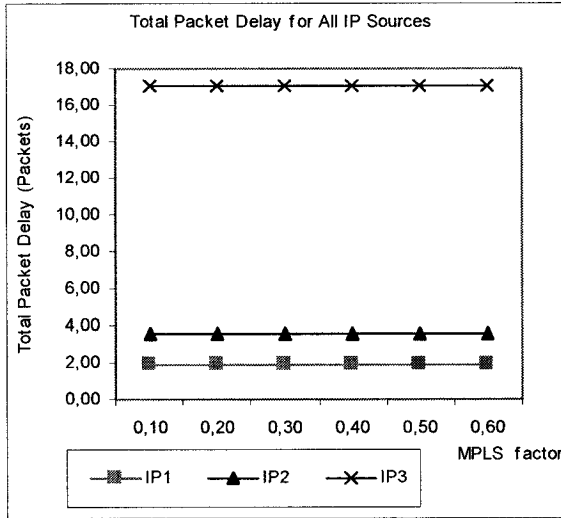
Fig. 6-30 Residual loss probability for IP sources versus IP factor (heterogeneous networks)



$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.4$

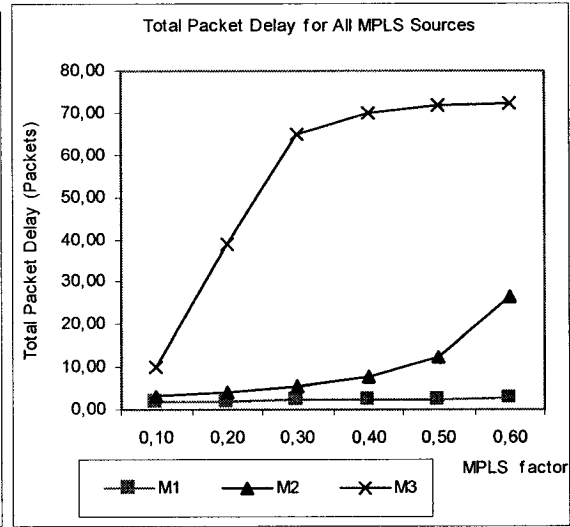
Fig. 6-31 Residual loss probability for MPLS sources versus IP factor (heterogeneous networks)





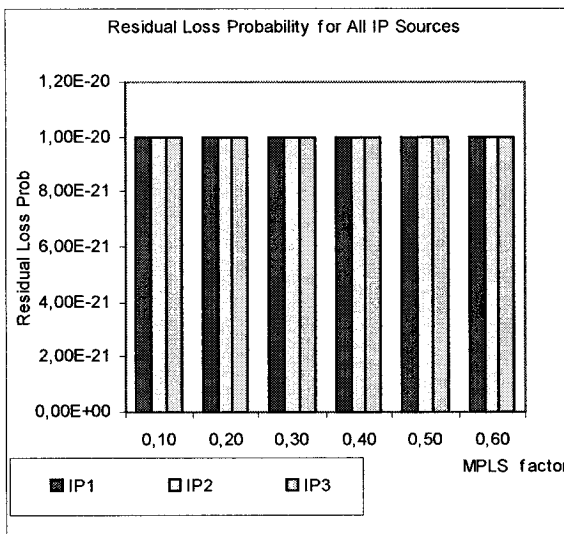
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.1$

Fig. 6-32 Total packet delay for IP sources versus MPLS factor (heterogeneous networks)



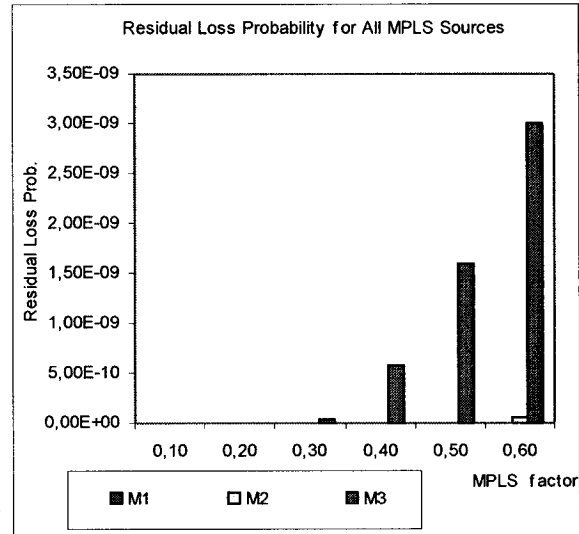
$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.1$

Fig. 6-33 Total packet delay for MPLS sources versus MPLS factor (heterogeneous networks)



$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.1$

Fig. 6-34 Residual loss probability for IP sources versus MPLS factor (heterogeneous networks)



$\alpha_1^1 = 0.25, \alpha_1^2 = 0.2, \alpha_1^3 = 0.15, \Delta = 0.1, D = 5,$   
 $B = 30, \xi_2 = 0.1, L = 500, z = 2, r = 223 / 255, \tau = 1.1$

Fig. 6-35 Residual loss probability for MPLS sources versus MPLS factor (heterogeneous networks)

## 6.4 Conclusions

The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). Each router may have different capabilities; for example one router could have the ability to correct errors (FEC) and use ARQ, one may use ARQ only and cannot correct errors, a third one may not have MPLS capability.

Section 6.2 tries to put an image for the expected future Internet and analyze the performance of this Internet when most of the current IP routers are replaced with MPLS routers. However, the replacement may not include all IP router and some IP routers will remain in this Internet and coexist with MPLS routers especially during the migration process. This would create a heterogeneous MPLS network. Section 6.2 compares the performance of this heterogeneous MPLS network with homogeneous IP and homogeneous MPLS networks using analysis tools. The results are general and are evaluated for a wide range of traffic values, priorities, etc for a complete binary tree. In addition to that a Fair Share Policy (FSP), which is a traffic policing mechanism, is proposed to ensure proper QoS.

are used in this comparison. The study found that the no FEC or ARQ mechanism (case1) is the best mechanism in terms of total packet delay for all IP, MPLS and heterogeneous sources. In addition to that, the hybrid FEC ARQ unicast mechanism is the best mechanism in terms of the residual packet loss probability for all IP, MPLS and heterogeneous sources. Also, we have found when MPLS multicast networks are mixed with some IP routers (heterogeneous network), it will perform less efficiently than homogeneous MPLS multicast networks. In addition to that when the IP and the

processing factors are small, the heterogeneous multicast network will perform less efficiently than IP homogeneous networks.

In section 6.3, a performance comparison between 4 types of multicast networks: homogeneous IP, homogenous MPLS, heterogeneous IP and heterogeneous MPLS is carried using analysis tools. In addition to that a Fair Share Policy (FSP), which is a traffic policing mechanism, is proposed to ensure proper QoS. Also, Differentiated Services and reliable FEC/ARQ with unicast repairs mechanism are used in this comparison. The study showed that when multicast networks are mixed with some DR routers (heterogeneous network), they will perform more efficiently than homogeneous multicast networks in terms of total packet delay and residual packet loss probability; and this is valid for both IP and MPLS multicast networks. This study suggests that as the number of DR routers in the multicast network increases, the network will have better performance in terms of total packet delay and residual packet loss probability.

In this chapter, we found that when the difference in packet processing time ( $\tau$ ) between IP and MPLS is high and when MPLS factor is small, IP multicast network will perform less efficiently than MPLS multicast network in terms of the total packet delay and the residual packet loss probability. However, when this difference in packet processing time is small IP performs very similar to MPLS. In addition to that when MPLS has higher arrival rate due to MPLS trees establishment control overhead and when the processing factor is small, IP would perform better than MPLS. We have found that the previous results are true for both homogeneous and heterogeneous IP or MPLS multicast networks.

Analysis results revealed that there is a noticeable improvement in QoS defined as the total packet delay and the residual packet loss probability for a higher priority traffic when MPLS multicasting replaces IP multicasting especially if MPLS factor is small and processing factor is large.

# **CHAPTER 7 SIMULATION OF FEC/AQR MULTICAST FOR DIFFSERV OVER MPLS AND IP PLATFORMS**

## **7.1 Introduction**

A hybrid FEC/ARQ strategy should be used where a combination of FEC for the most frequent error patterns, together with error detection and retransmission for the less likely error patterns is more efficient than ARQ alone. In this case, when FEC fails to correct errors at the receiver the receiver sends a NAK to the sender to retransmit the data in error. This hybrid FEC/ARQ strategy clearly carries the potential for improving throughput in two-way systems subject to a high channel error rate. In this chapter, we will evaluate the QoS performance with multicast repairs mechanism only.

In this chapter, we compare QoS performance of IP and MPLS multicasting, given their particular constraints [113-114]. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. We present a fair share policy and by taking the above constraints into consideration, we evaluate the QoS performance for a typical binary tree in the two cases of IP and MPLS multicasting. We also consider Differentiated Services; i.e. traffics with different priority classes when reliable FEC/ARQ multicast is used. Simulation programs will be used to evaluate our fair share policy (FSP) for different network scenarios. The rest of the chapter is structured as follows: section 7.2 explains and describes and the simulation model underlying the fair share policy used in this chapter. Section 7.3 introduces the performance criteria used to evaluate the FSP for both homogeneous and heterogeneous

multicast cases. Section 7.4 will carry a comprehensive comparison between IP and MPLS homogeneous multicast networks when DiffServ and FEC/ARQ reliable multicast are adopted. However, section 7.5 will carry a similar comparison but for heterogeneous MPLS networks. Section 7.6 compares some interesting results obtained using both analysis and simulation. Section 7.7 explains how the validity of the simulation programs is achieved. Finally, conclusions are summarized in section 7.8.

## **7.2 The Simulation Model Underlying the Fair Share Policy (FSP)**

FSP is not a call admission rather it is a traffic policing mechanism. In FSP, packets are discarded in case of congestion differently at each queue according to source priority and the maximum number in the queue; i.e. the source with higher priority will experience less packet discarding than sources with lower priorities. Moreover, FSP guarantees fairness among flows having the same priority (i.e., required QoS) in buffer space allocated to lower priority traffic is larger; thus leading to less packet discard [113-114].

To evaluate and to gain in-depth insights into our performance model, we used simulation to evaluate and compare QoS performance between IP and MPLS multicast networks when Hybrid FEC/AQR and DiffServ are adopted. Our simulation model is shown in Fig. 7-1.

In this model, a typical IP or MPLS router and our FSP traffic policing mechanism process three independent sources corresponding to different input traffic classes. Source

1 is assigned the highest priority, then source 2 and finally source 3. For this model, the enforcement is assumed to occur at the router (node) according to Fair Share Policy.

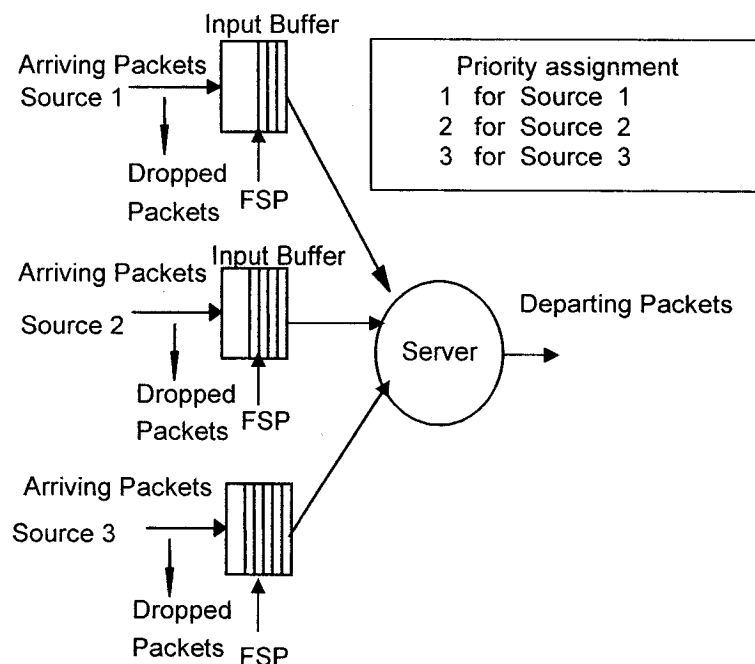


Fig. 7-1 The Router's Simulation Model

### 7.2.1 Simulation Model Assumptions

The following assumptions are used:

- 1- For the convenience of simulation and for its memory less property, the arrival of packets is assumed to follow a Poisson process with exponentially distributed mean inter-arrival times of packets in the queue of each priority class. The mean inter-arrival time of priority  $p$  packets can be found as:

$1/\lambda_p = \Delta T / \alpha_p$  where  $\Delta T$  is the service time which equals to  $(1/250000)$  for an output line with 1Gb/s speed.  $\alpha_p$  is priority  $p$  arrival probability.

- 2- FSP uses non pre-emptive priority queuing and FIFO for the same priority packets.

- 3- The arrival probabilities are  $\alpha_1, \alpha_2$  and  $\alpha_3$  for each source respectively.
- 4- Service probabilities for different queues are  $\beta_1, \beta_2$  and  $\beta_3$  for each source respectively, which take the priorities into account; i.e. during any packet time server is available only for one class as will follow shortly.
- 5- Average queue sizes are  $E_1(n), E_2(n)$  and  $E_3(n)$  for each source respectively.
- 6- Maximum buffer sizes are  $\max_1, \max_2$  and  $\max_3$  for each source respectively.
- 7- Total system buffer size is fixed i.e. ,  $B = \max_1 + \max_2 + \max_3$  where  $\max_p$  and  $p = 1, 2, 3$  is calculated as:  $\max_p = \frac{Pr_p}{\sum_p Pr_p} * B$  where  $Pr_p$  is source  $p$  priority.
- 8- All of MPLS or IP routers on the subject Internet are identical in providing source and traffic conditions.
- 9- All packets are of the same length and consist of 500 bytes.
- 10- A complete binary multicast tree is used, where each parent router has two children routers until we reach leafs.
- 11- Routing and rerouting are not part of this research
- 12- Acknowledgements are never lost.

Fig. 7-1 explains the main components of the analytical model for a typical router (IP or MPLS). FSP sets the following rule: low priority users will generally have higher buffer occupancy  $E(n)$  which provides the low priority traffic with more space as needed. The classifier, which is not shown, aggregates all users' traffics of a certain priority and sends them into the same priority queue.

## 7.2.2 Input Buffer Characteristics

The parameters of the model are as follows: each source  $p$  ( $p=1, 2,$  or  $3$ ) has limited input buffer. This input buffer service discipline is a FIFO queue. A packet arriving at a



particular input buffer goes directly to the output buffer provided that the input buffer is empty (i.e. no previous packets waiting in that input buffer) and there are no packets waiting from higher priority traffic. Otherwise, the packet will be waiting in the input buffer until the preceding packets are served (if there are any). A packet arriving at a particular buffer is rejected (dropped) if the corresponding buffer is full.

Packets from source 1 are always served first in order to give highest priority source the best service probability. Packets from source 2 will be served only when the buffer corresponding to source 1 (which has higher priority) is empty. Finally, packets from source 3 will be served only when the buffers corresponding to source 1 and source 2 (which have higher priority) are all empty.

### 7.2.3 Server (Sender) Characteristics

Packets in the input buffers are statistically multiplexed into an infinite output buffer. The output channel transmits packets from the output buffer at a fixed bandwidth rate, which is much greater than the sum of the average arrival rates of the input sources. In the simulation programs, an output link rate of 1Gbit/s is used, which equals to 250,000 packets/s for 500 bytes packet length. For not affecting the QoS performance of a FSP, we assume the availability of the server for all HOL packets that are allowed to be transmitted to the output buffer by FSP.

### 7.2.4 IP and MPLS Source Arrivals

For IP based networks, the source arrival probability  $\alpha$  is actually a composite one; for instance  $\alpha_1$  can be written as:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad (7-1)$$

where  $\xi_1$  is the IP control overhead factor (or IP factor).

$\alpha_1^1$  is the intrinsic arrival probability at the application layer (on top of IP layer),  $\alpha_1^2$  is the extra arrival probability due to IP control overhead which is used to establish the IP multicast tree.

Similarly for MPLS based networks,  $\alpha_1$  can be written as:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3$ , where  $\alpha_1^1$  and  $\alpha_1^2$  are the same as in the case of IP networks;  $\alpha_1^3$  is the extra arrival probability due to MPLS control overhead which is used to establish the MPLS multicast paths or tree.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2)\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad \xi_2 = \frac{\alpha_1^3}{\alpha_1^1} \quad (7-2)$$

Where  $\xi_2$  is the MPLS control overhead factor (or MPLS factor).

### 7.2.5 Interleaving

In this work, interleaving is used in order to break byte burst losses and efficiently turn them independent random byte losses at the source and destination [125]. Interleaving is done at the Rendezvous Point where original source packets are interleaved together to create one interleaved packet. This interleaved packet is sent to multicast network where the reverse process, i.e. the original packet is restored from these interleaved packets.

## 7.3 Performance Measures

In this chapter, we will use the following measures to evaluate and compare the QoS performance between IP and MPLS multicast networks when hybrid FEC/ARQ and DiffServ are adopted:

**1- Total Packet Delay for Priority Traffic p (p=1,2,3).**

$$D_{pTotal} = \sum_{j=1}^D D_{j,p} \quad (7-3)$$

Where D is the number of routers in the longest path (Depth),  $D_{j,p}$  is router's j average packet delay for priority traffic p, which is given as:

$$D_{j,p} = \frac{\sum_{i=1}^T D_{i,j,p}}{T} \quad (7-4)$$

Where  $D_{i,j,p}$  is the average packet delay in router's j queue for iteration i and for priority traffic p. T is the total number of iterations (multicast packets sent).

Then, the average packet delay per a router in the multicast tree for priority traffic p (p=1,2,3) can be given as:

$$\bar{D}_p = \frac{D_{pTotal}}{D} \quad (7-5)$$

**2- Packet Delay Jitter for Priority Traffic p.**

$$\sigma_{xp} = \sqrt{\frac{\sum_{j=1}^D (\bar{D}_p - D_{j,p})^2}{D - 1}} \quad (7-6)$$

**3- Residual Packet Loss Probability for Priority Traffic p**

$$Ploss_p = (\sum_{j=1}^N PL_{j,p}) / N \quad (7-7)$$

Where N is the total number of routers in the multicast network,  $PL_{j,p}$  is router's j average packet loss probability for priority traffic p, which is given as:

$$PL_{j,p} = \frac{\sum_{i=1}^T PL_{i,j,p}}{T} \quad (7-8)$$

Where  $PL_{i,j,p}$  is the average packet loss probability in router's  $j$  queue for iteration  $i$  and for priority traffic  $p$ .  $T$  is the total number of iterations (multicast packets sent).

## **7.4 Homogeneous IP/MPLS Multicast Networks**

### **7.4.1 Simulation Programs**

Simulation is used to evaluate and compare the QoS performance between IP and MPLS platforms in the presence of FSP. Popularity of the simulation tool stems from the fact simulation makes it possible to systematically study a network to a desired level of detail, when exact analysis is either not possible or too expensive, or whenever an approximate analytic solution is not adequate.

### **7.4.2 Description of the Simulation Programs**

The simulation flowcharts for our multicast model are shown in Fig. 7-2. Thesis programs are based on the asynchronous timing approach of event-driven techniques [115,128,129]. This technique is chosen due to its suitability for simulation of computer network events and because the dependency relation characteristics of the technique, allow easier validation of the simulation models.

### **7.4.3 The Simulation Modules**

Our simulation programs are discrete event driven simulation programs. These programs are part of the thesis development and are created using a high-level programming language. The programs are highly modular and flexible allowing various

input source characteristics and various policing mechanism characteristics to be used. The flowchart of our simulation program is shown in Fig. 7-2. The simulation programs consist of the following main modules:

- 1- Initialization Module
- 2- Event Generator Module
- 3- Event Scheduler Module
- 4- Error Detection/Correction Module

The Event generator module performs the following functions:

- Generation of the packet arrival events depending upon the input source characteristics (Poisson distribution in this case).
- Generation of the departure events at slotted times depending upon the bandwidth of the output link. Each slot can carry one packet at a time.

The Event scheduler scans through the event list and accordingly schedules the next event for execution.

The Error Detection/Correction Module responsible for implementing FEC/ARQ operation.

#### **7.4.4 FEC/ARQ Operation**

Reed-Solomon code RS (n,k) is used with  $n=255$ ,  $k=223$  and 8-bits symbols. Every code word contains 255 code word bytes, of which 223 bytes are data and 32 bytes are parity. The procedure can be summarized as follows:

- 1- Keep a counter for each source packet to the number of lost bytes (erasures) and the number of bytes in error.

2- At the reception of a packet at the router; the software checks if there is a room to buffer this packet. If there is no room, then a packet overflow will occur and this packet would be discarded. Since the discarded packet is actually an interleaved packet, therefore only one byte from each source packet (before interleaving) will be lost. Increment each source packet counter by 1.

3- Since the purpose of this work is not details of FEC encoding/decoding, we emulate the FEC operation by calling an FEC function (for each byte in the packet) that would generate a random number  $U(0,1)$ ; then we compare this random number with a predefined number that defines the FEC correction probability. If this random number is smaller than FEC number, then the byte is correct. Otherwise, the byte is in error, then increment corresponding source packet counter by 1.

To illustrate, suppose the FEC number that represents FEC ability to correct errors is 0.9. After calling  $U(0,1)$  and obtain a random number of 0.95; since the obtained random number (0.95) is greater than FEC number (0.9), then the byte is in error and the corresponding source packet counter of this byte is incremented by 1. Now assume the random number is 0.7 then the byte is considered correct. In thesis programs, FEC ability of 0.98 is used.

4- For each source packet, check its counter of lost bytes and bytes in error. If this counter is greater than the correction capability of the FEC operation (FEC distance), then the source packet is considered lost and a NAK for this packet is sent to the sender or Rendezvous Point to repeat sending the lost source packet. This is considered as one ARQ trial. If any router fails to receive the packet after all ARQ trials, the packet is

considered residually lost. In thesis simulation programs, ARQ trials of two times are used.

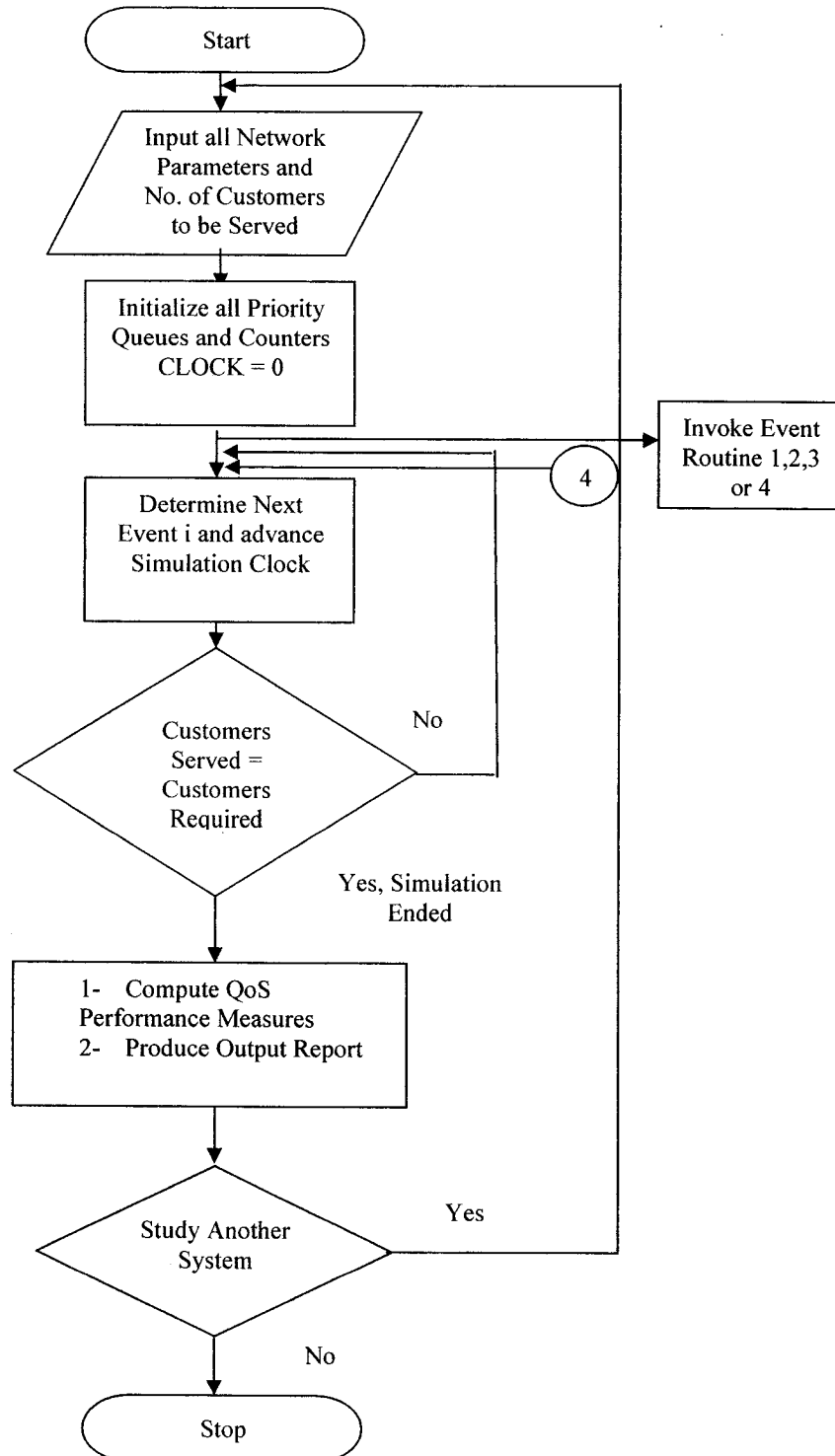


Fig. 7-2 (a) Flowchart of the Simulation Program Part A (Initialization and Main Module)

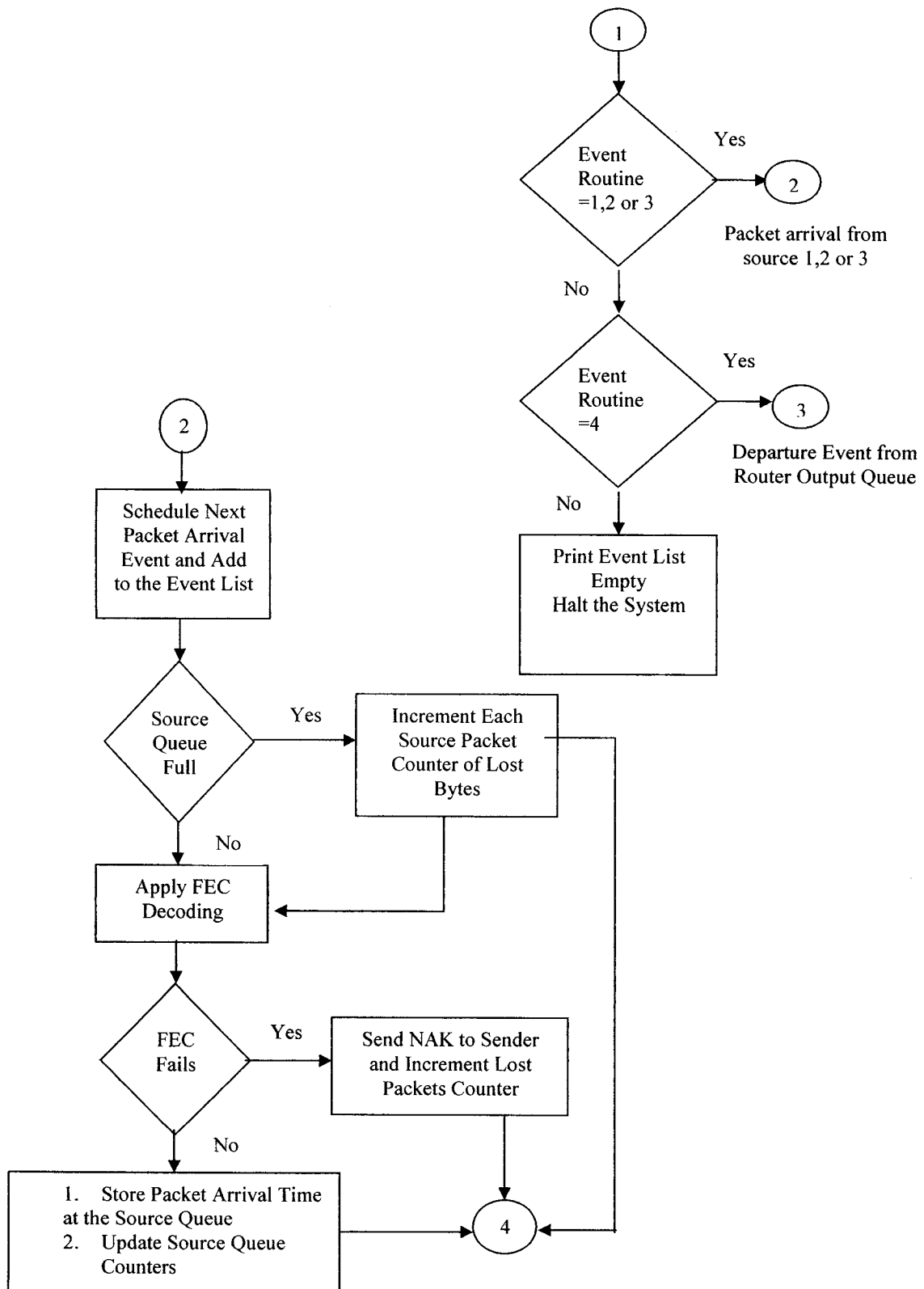


Fig. 7-2 (b) Flowchart of the Simulation Program Part B (Event Generator and FEC/ARQ Operation)



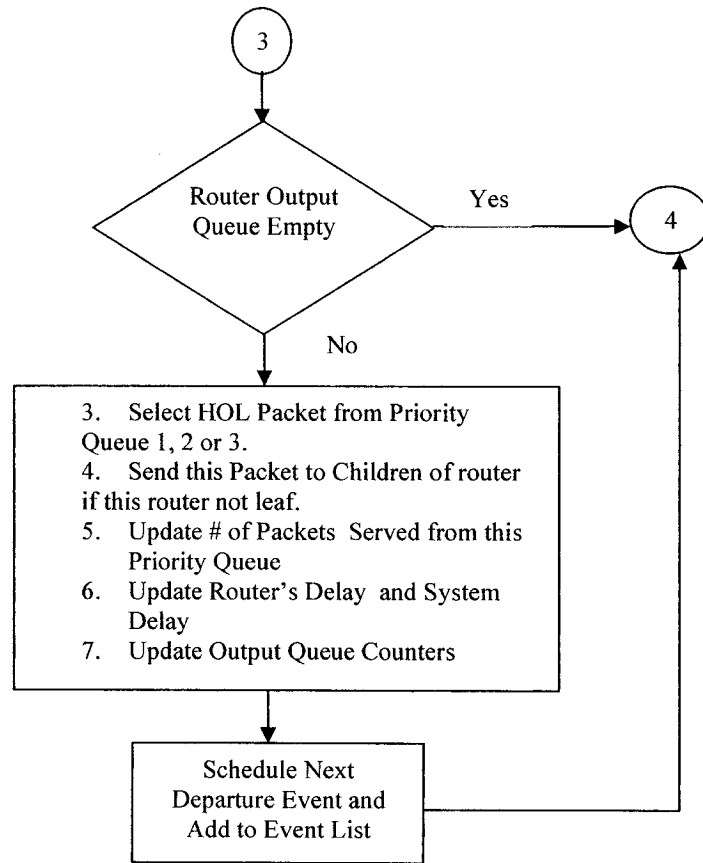


Fig. 7-2 (c) Flowchart of the Simulation Program Part C (Event Scheduler)

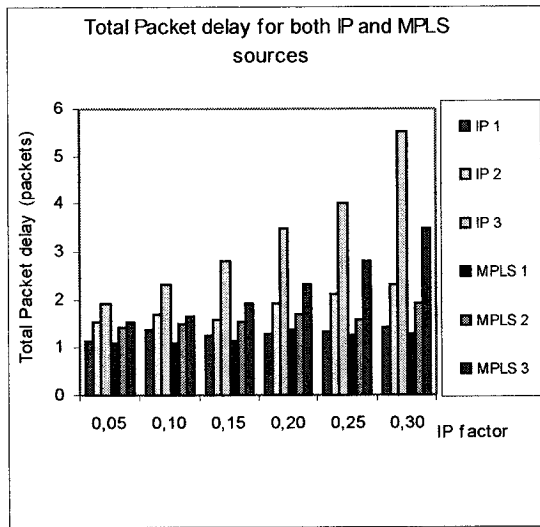
### 7.4.5 Simulation Results

Figs. 7-3 to 7-11 show the performance comparisons between IP sources and MPLS sources in the multicast tree when hybrid FEC/ARQ with multicast repairs is applied. Fig. 7-3 shows the total packet delay for all sources for both IP and MPLS versus IP factor for small processing factor ( $\tau$ ). It shows that IP and MPLS have almost the same total packet delay, except a small difference for source 3. Fig. 7-4 shows the delay jitter for all sources for both IP and MPLS versus IP factor for small processing factor. In addition, it shows that both IP and MPLS have very much the same delay jitter, except a

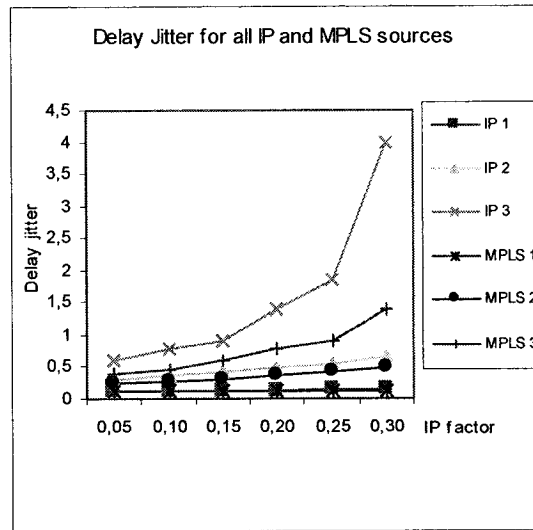
small difference for source 3; and as IP factor increases, the difference becomes even smaller. Fig. 7-5 shows the residual packet loss probability for all sources for both IP and MPLS versus IP factor for small processing factor. In addition, it shows that IP and MPLS sources have very same residual loss probability (almost zero).

However, Figs. 7-6, 7-7 and 7-8 show that when the processing factor ( $\tau$ ) increases MPLS will have superiority over IP in terms of the total packet delay, delay jitter and the residual packet loss probability. As shown in Figs. 7-6 and 7-8, the total packet delay and the residual packet loss probability in case of MPLS are less than IP for all sources and these differences are clear for low priority sources 2 and 3. Fig. 7-7 shows that the delay jitter in the case of MPLS is less than IP for all sources except for MPLS source 3 which starts smaller the IP source 3 and it continues to increase with the increase of IP factor. This means when the difference in packet processing ( $\tau$ ) between MPLS and IP increases, MPLS in general will be better.

In Figs. 7-3 to 7-8 MPLS factor was constant and relatively small; explaining why MPLS performance was better or very similar to IP performance. However, in the following figures we will study the effects of MPLS factor on MPLS performance. Figs. 7-9, 7-10 and 7-11 show that IP will be superior over MPLS when MPLS factor increases. As shown in Figs. 7-9, 7-10 and 7-11 the total packet delay, the delay jitter and the residual packet loss probability in the case of IP (which are constant) are less than MPLS. This means when the extra arrival rate due MPLS control overhead used to establish MPLS multicast paths or tree increases, IP will be perform better especially when the intrinsic traffics increase.



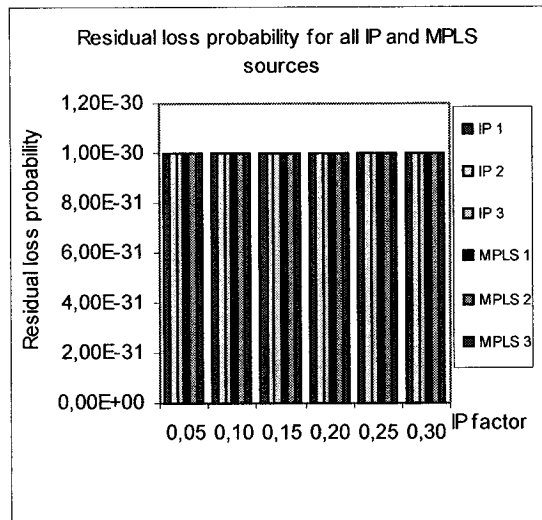
$$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12, \\ D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2$$



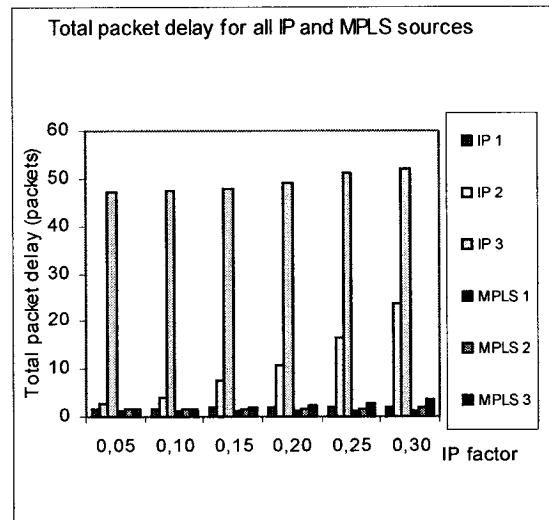
$$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12, \\ D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2$$

Fig. 7-3 Total packet delay versus IP factor (small  $\tau$ )

Fig. 7-4 Delay jitter versus IP factor (small  $\tau$ )



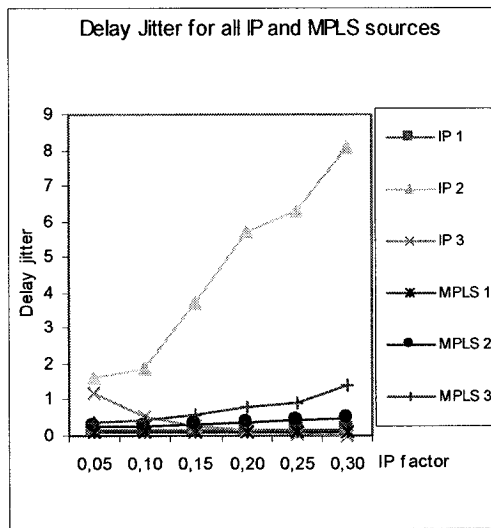
$$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12, \\ D = 4, B = 30, \xi_2 = 0.1, \tau = 1.2$$



$$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12, \\ D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$$

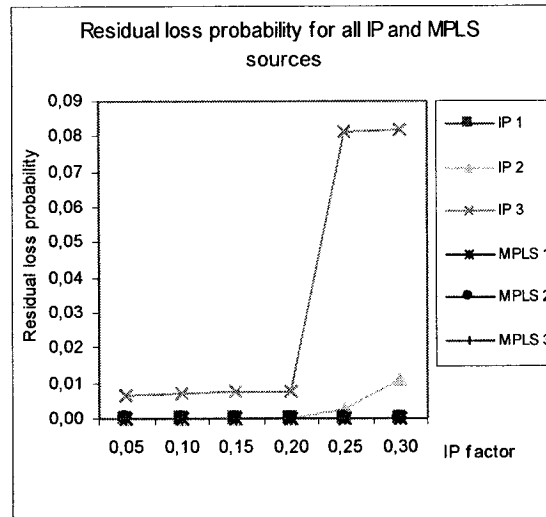
Fig. 7-5 Residual loss probability versus IP factor (small  $\tau$ )

Fig. 7-6 Total packet delay versus IP factor (large  $\tau$ )



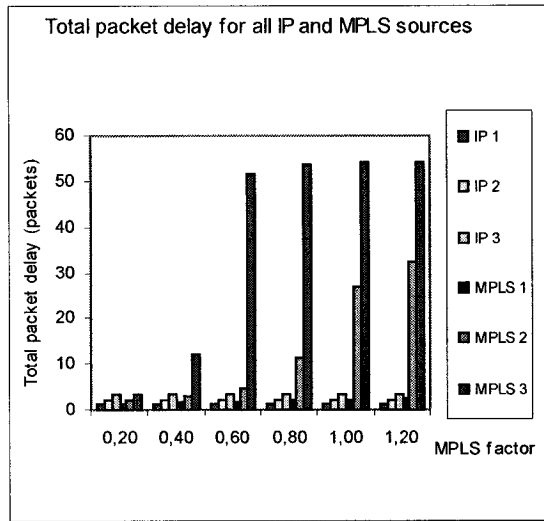
$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$

Fig. 7-7 Delay jitter versus IP factor (large  $\tau$ )



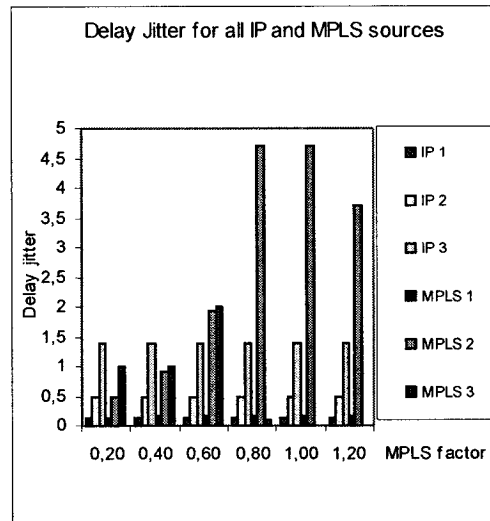
$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12,$   
 $D = 4, B = 30, \xi_2 = 0.1, \tau = 1.8$

Fig. 7-8 Residual loss probability versus IP factor (large  $\tau$ )



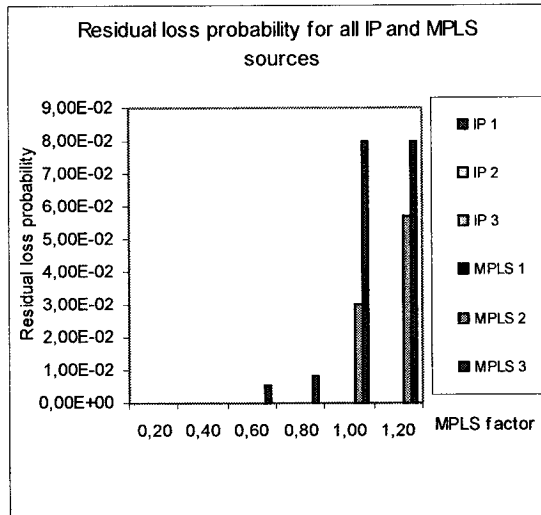
$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12,$   
 $D = 4, B = 30, \xi_1 = 0.1, \tau = 1.2$

Fig. 7-9 Total packet delay versus MPLS factor



$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12,$   
 $D = 4, B = 30, \xi_1 = 0.1, \tau = 1.2$

Fig. 7-10 Delay jitter versus MPLS factor



$$\alpha_1^1 = 0.23, \alpha_1^2 = 0.17, \alpha_1^3 = 0.12,$$

$$D = 4, B = 30, \xi_1 = 0.1, \tau = 1.2$$

Fig. 7-11 Residual loss probability versus MPLS factor

## 7.5 Heterogeneous MPLS Multicast Networks

In this section, three different types of multicast networks will be compared. In the homogeneous IP multicast network, all routers are IP routers. In the homogeneous MPLS multicast network, all routers are MPLS routers while in the heterogeneous MPLS multicast network, the network is assumed to be MPLS network but still having some IP routers. This is a practical situation that happens during the migration process from all IP routers to all MPLS routers networking. The number and location of these IP routers in this MPLS network will create the different situations as table 6-1 shows. Each different situation may create up to four types of routers in the MPLS heterogeneous network as would be explained in this section. In this type of network, we can have no IP router in the network (homogeneous MPLS case), 1 IP router in the network, 2 IP routers in the network, or 3 IP routers in the network.

These IP routers can be located anywhere in the network (in our example a complete 31 nodes binary tree is taken) except the root, which is the sender or the Rendezvous Point router. The root is assumed always to be an MPLS router. In any case, there will be up to four types of routers:

- 1- IP (type1) router, which is a regular IP router.
- 2- ME (type 2) router, which is an MPLS router with extra processing due to more packet processing is needed at the MPLS router because the downstream router is an IP router.
- 3- EI (type 3) router, which is either an egress or ingress router with extra processing due to the overhead of tunnel establishment and maintenance and also due to more packet processing is needed because of the IP routers which reside in between EI routers.
- 4- M (type 4) router, which is a regular MPLS router.

### 7.5.1 Different Source Arrivals

For IP based networks, the source arrival probability  $\alpha$  is actually a composite one; for instance  $\alpha_1$  can be written as:

$$\alpha_1 = \tau\alpha_1^1 + \xi_1\alpha_1^1 \quad \xi_1 = \frac{\alpha_1^2}{\alpha_1^1} \quad (7-9)$$

Similarly for regular **MPLS routers** (type 4),  $\alpha_1$  can be written as:

$$\alpha_1 = (1 + \xi_1 + \xi_2)\alpha_1^1 \quad (7-10)$$

For **ME routers** (type 2),  $\alpha_1$  can be written as:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^4$ , where  $\alpha_1^1$ ,  $\alpha_1^2$  and  $\alpha_1^3$  are the same as in the case of regular MPLS router;  $\alpha_1^4$  is the extra arrival probability due to the overhead on the MPLS router because the downstream router is an IP router.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2 + \xi_3)\alpha_1^1 \quad (7-11)$$

Where  $\xi_3$  is the ME factor and  $\xi_3 = \frac{\alpha_1^4}{\alpha_1^1}$

For **EI routers** (type 3),  $\alpha_1$  can be written as:

$\alpha_1 = \alpha_1^1 + \alpha_1^2 + \alpha_1^3 + \alpha_1^5$ , where  $\alpha_1^1$ ,  $\alpha_1^2$  and  $\alpha_1^3$  are the same as in the case of regular MPLS router;  $\alpha_1^5$  is the extra arrival probability due to the overhead of tunnel establishment and maintenance and also due to the fact that more packet processing is needed because of the IP routers which reside in between EI routers.  $\alpha_1$  can be rewritten in terms of  $\alpha_1^1$  as:

$$\alpha_1 = (1 + \xi_1 + \xi_2 + \xi_4) \alpha_1^1 \quad (7-12)$$

Where  $\xi_4$  is the EI factor and  $\xi_4 = \frac{\alpha_1^5}{\alpha_1^1}$

The same simulation model, simulation assumptions, FSP, input buffer characteristics, interleaving and server characteristics that were introduced in sections 7.2 and 7.4 will be used here. In addition to that, the same criteria that were defined in section 7.3 will be used to carry the comparison between homogeneous IP networks, homogeneous MPLS networks and heterogeneous MPLS networks when DiffServ and reliable FEC/ARQ with multicast repairs are adopted. However, for heterogeneous MPLS networks, there would be extra subprograms that are used for heterogeneous case. These subprograms will be introduced in the next subsection.

### 7.5.2 Simulation Programs

In addition to the simulation programs that were used in section 7.4 (see Fig. 7-2), extra subprograms would be used for the heterogeneous MPLS multicast network case. The simulation programs consist of the following main modules:

1- Initialization Module

2- Event Generator Module

3- Event Scheduler Module

4- Error Detection/Correction Module

5- Heterogeneous MPLS Network Configuration Module

The first four modules are shown in Fig. 7-2, while the fifth one is shown in Fig. 7-12

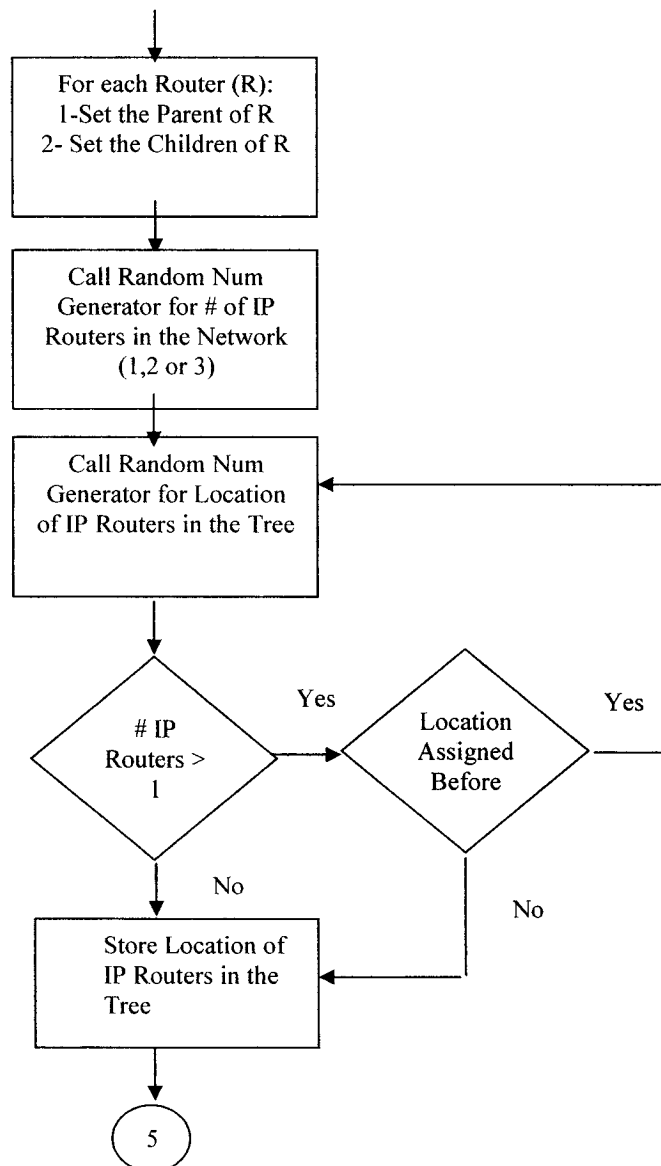


Fig. 7-12 (a)



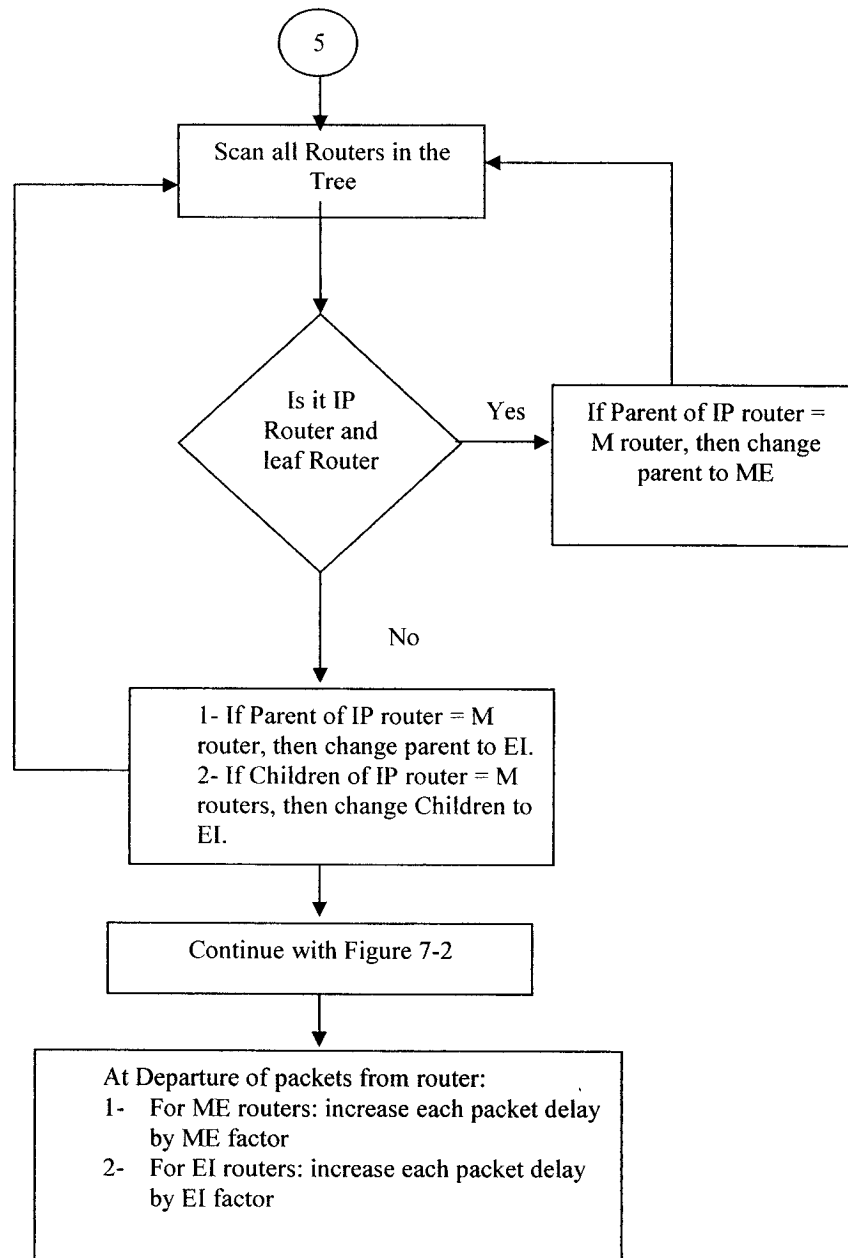


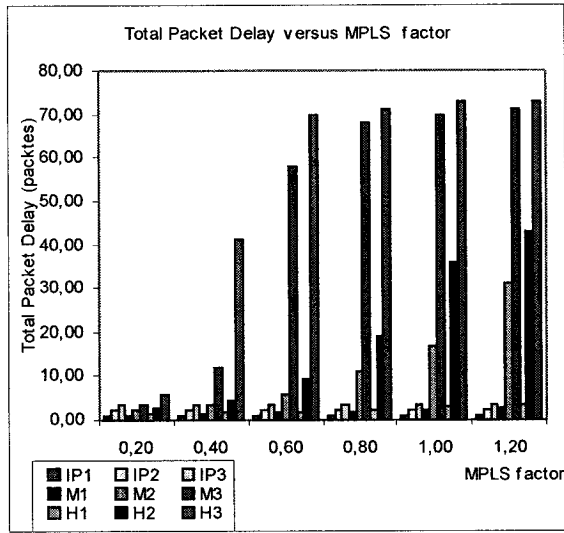
Fig. 7-12 (b)

Fig. 7-12 Flowchart of heterogeneous MPLS network configuration module

In this module and as shown in Fig. 7-12 (a), the number of IP routers (could be 1, 2, or 3) is first generated using a random number generator. Then the locations of these IP routers in the MPLS multicast tree are also generated randomly provided that two or more IP routers cannot be assigned the same location. Fig. 7-12 (b) shows and illustrates how the heterogeneous MPLS multicast tree would be configured according to the number and locations of IP routers. After, configuring the heterogeneous MPLS multicast tree, flowchart of Fig. 7-2 would be used. However, at the departure of packets from each router, an extra processing packet delay of ME factor ( $\xi_3=0.05$ ) or EI factor ( $\xi_4=0.1$ ) will be added if the router type found to be ME router or EI router respectively.

### **7.5.3 Simulation Results**

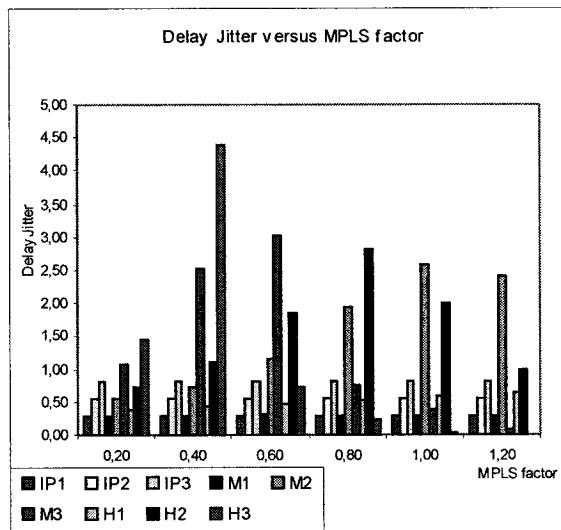
Figs. 7-13 to 7-15 show the performance comparisons between the three types of multicast networks when hybrid FEC/ARQ with multicast repairs mechanism is applied. These figures show the performance comparisons between homogeneous IP network (with each router has IP1, IP2 and IP3 sources), homogeneous MPLS network (with each router has M1, M2 and M3 sources) and heterogeneous MPLS network (with each router has H1, H2 and H3 sources). Fig. 7-13 to 7-15 show the total packet delay, delay jitter, and the residual packet loss probability for all sources versus MPLS factor. These figures show that IP sources will have the best performance in terms of total packet delay, delay jitter and residual packet loss probability; on the other hand, the heterogeneous sources will have the worst performance.



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$$

$$D = 5, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$$

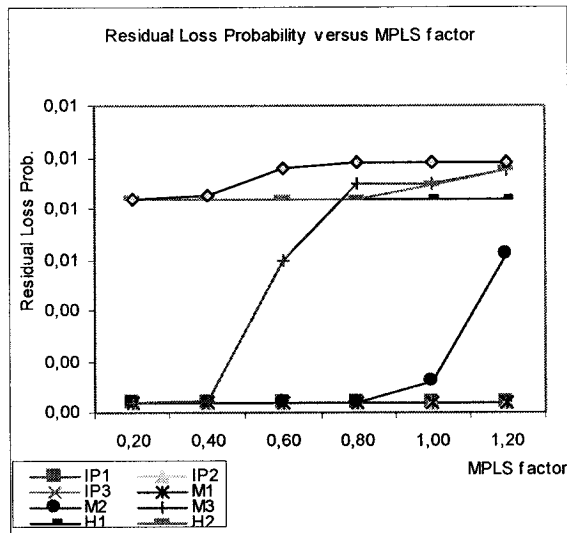
Fig. 7-13 Total packet delay for all sources versus MPLS factor



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$$

$$D = 5, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$$

Fig. 7-14 Delay jitter for all sources versus MPLS factor



$$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$$

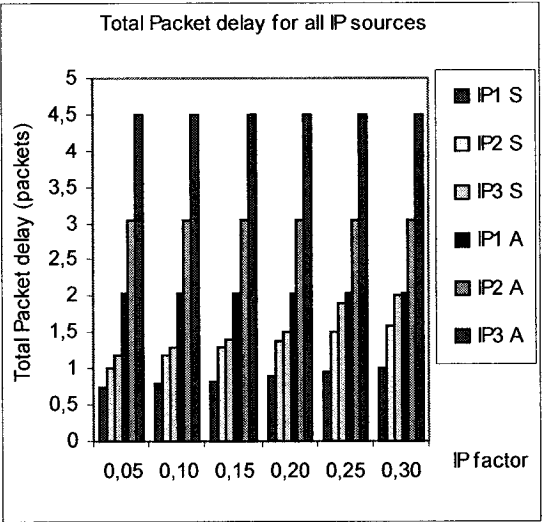
$$D = 5, B = 30, \xi_1 = 0.2, \tau = 1.2, L = 500$$

Fig. 7-15 Residual loss probability for all sources versus MPLS factor

## 7.6 Comparison of Analysis and Simulation Results

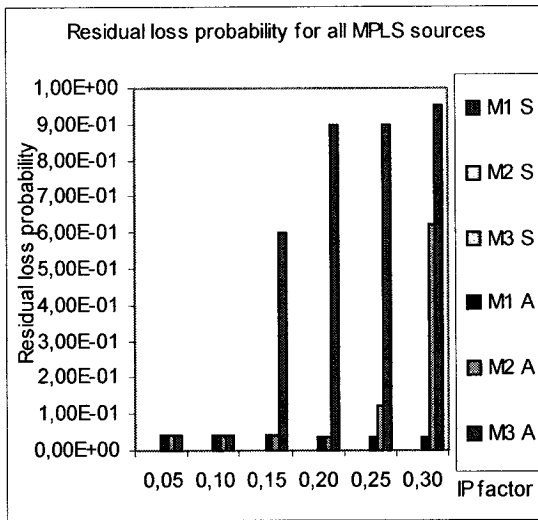
Figs. 7-16 to 7-18 show the comparison between analysis and simulation results for the three types of networks. Fig. 7-16 shows the performance comparison in terms of total packet delay between homogeneous IP simulation sources (IP1S, IP2S, and IP3S) and homogeneous IP analysis sources (IP1A, IP2A and IP3A). Fig. 7-17 shows the performance comparison in terms of the residual loss probability between homogeneous MPLS simulation sources (M1S, M2S, and M3S) and homogeneous MPLS analysis sources (M1A, M2A and M3A). Fig. 7-18 shows the performance comparison in terms of the residual loss probability between heterogeneous simulation sources (H1S, H2S, and H3S) and heterogeneous analysis sources (H1A, H2A and H3A). In all the mentioned figures, one may notice that simulation results have the same tendency as the analysis

results. In addition to that, simulation results are very close to analysis results. Fig. 7-17 shows that the residual loss probability for all simulation MPLS homogeneous sources is very small (almost 0). However, Fig. 7-17 shows that the residual loss probability for all analysis MPLS homogeneous sources starts with small value when the arrival probability is small, and increases as the arrival probability increases until it reaches 0.9 for source 3, which is the lowest priority source. That is because many assumptions were made in the case of analysis and some of these assumptions were built on a worst-case scenario.



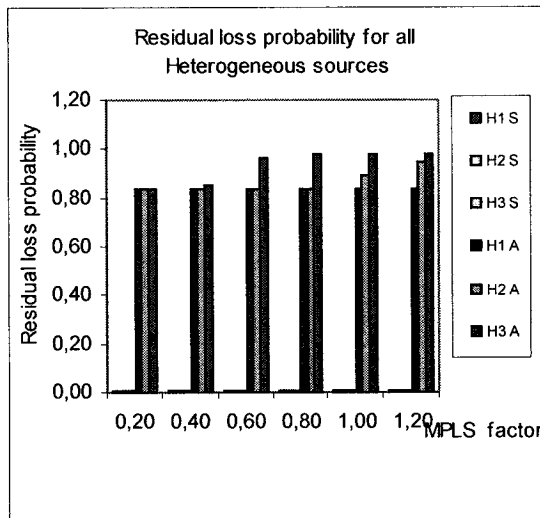
$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_2 = 0.1, \tau = 1.2, L = 500, z = 2$

Fig. 7-16 Total packet delay for all IP sources (homogeneous)



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_2 = 0.1, \tau = 1.2, L = 500, z = 2$

Fig. 7-17 Residual loss probability for all MPLS sources (homogeneous)



$\alpha_1^1 = 0.2, \alpha_1^2 = 0.15, \alpha_1^3 = 0.1, r = 223/255$   
 $D = 5, B = 30, \xi_1 = 0.1, \xi_3 = 0.05,$   
 $\xi_4 = 0.1, \tau = 1.2, L = 500, z = 2$

Fig. 7-18 Residual loss probability for all MPLS sources (heterogeneous)

## 7.7 Validity of the Simulation Programs

The validity of thesis simulation programs is confirmed with respect to two things:

- 1- A large number of packets is served by system for a single run.
- 2- In order to achieve an interval of confidence of 95% with 4% or less error, each program is kept running for n times to obtain a single data. According to the rule

$$[123]: \quad n = (Z_{\alpha/2} \sigma / e)^2 \quad (7-13)$$

Where  $\sigma$  is the sample variance.

In order to find n, we should first find  $\sigma$ . At the beginning we start with a small value of n say (n=3). After that, we approximate  $\sigma$  using the following:

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (\bar{y} - y_i)^2}{n-1}} \quad \text{Where } \bar{y} \text{ and } y_i \text{ are the average and the individual}$$

performance measure (delay or residual loss probability ,..etc). Then we plug the obtained  $\sigma$  into equation (7-13) to calculate the required number of runs n. After that, we compare the calculated n with the assumed n. If the calculated n in equation (7-13) is equal or less than the starting n, we stop, otherwise we iterate.

Where  $0 < \alpha < 1$  and  $(1-\alpha)100\%$  is called the confidence interval or the degree of confidence. In our case  $\alpha = 0.05$ .

From tables [123]  $Z_{\alpha/2}$  which is the z-value leaving an area of  $\alpha/2$  to the right can be found to be equal 1.96. For an error  $e=4\%$ , and for the used performance criteria, the number of runs was found to be at most  $n=15$ .

## 7.8 Conclusions

In this chapter, a performance comparison between 3 types of multicast networks: homogeneous IP, homogenous MPLS, and heterogeneous MPLS is carried using

simulation programs. Also, a Fair Share Policy (FSP), which is a traffic policing mechanism, is proposed to ensure proper QoS. In addition, Differentiated Services and reliable FEC/ARQ with multicast repairs were used in this comparison.

In this chapter, we found that when the difference in packet processing time between IP and MPLS is high and when MPLS factor is small, IP multicast would perform less efficiently than MPLS in terms of the total packet delay, delay jitter and the residual packet loss probability. However, when this difference in packet processing time is small IP performs very similar to MPLS. In addition to that, when MPLS has higher arrival rate due to MPLS trees establishment control overhead and when the processing factor is small, IP would perform better than MPLS.

The study found that when MPLS multicast networks are mixed with some IP routers (heterogeneous network), it will perform less efficiently than homogeneous MPLS multicast networks. In addition to that, when the IP factor is small, the heterogeneous multicast network will perform less efficiently than IP homogeneous networks.

Simulation results revealed that there is a noticeable improvement in QoS defined as the total packet delay, delay jitter and the residual packet loss probability for higher priority traffic when MPLS multicasting replaces IP multicasting especially if MPLS factor is small and processing factor is large. However, the difference between the two QoS provided by MPLS and IP becomes minimal for low priority traffic.

In addition to that, the study finds that using hybrid FEC/ARQ would have positive impact on the residual packet loss probability. However, it may cause an increment on the total packet delay due to ARQ operation especially when the network is noisy and when FEC decoder fails to correct errors.



# **CHAPTER 8 THESIS CONCLUSIONS AND FUTURE WORK**

## **8.1 Thesis Conclusions**

In this thesis, we presented a new fair share policy (FSP) that utilizes Differentiated Services traffic to solve the problems of QoS and congestion control when multicasting is used. FSP is not a call admission rather it is a traffic policing mechanism. In FSP, packets are discarded in case of congestion differently at each queue according to source priority and the maximum number in the queue; i.e. the source with higher priority say real time voice and video will experience less packet discarding than sources with lower priorities. Moreover, FSP guarantees fairness among flows having the same priority (i.e., required QoS) in buffer space allocated to lower priority traffic say email or web browsing is larger; thus leading to less packet discard. This mechanism also replaces the hybrid MPLS/RSVP model and solves its problems related to multicasting. A comparison of QoS performance of IP and MPLS multicasting, given their particular constraints was carried out in both homogeneous and heterogeneous networks. In regular IP multicasting only overhead pertaining to IP multicast tree should be established, while in MPLS multicasting we have to add also the corresponding MPLS multicast tree establishment times and control packets. In this thesis the QoS performance was evaluated using different performance measures such as total packet delay, residual packet loss probability and delay jitter of DiffServ classes (traffics with different priority classes) for both MPLS and IP platforms. In order to achieve the required QoS, different techniques of reliable multicasting were adapted, such as Forward Error Correction (FEC) or Automatic

Repeat Request (ARQ) or Hybrid FEC/ARQ with multicast and unicast repairs mechanisms so as to mitigate the effect of errors as well as packet loss. This reliable multicast was used for both IP and MPLS platforms with Diffserv.

Analytical and simulation models were suggested and used. A model represents a typical IP or MPLS router and FSP traffic policing mechanism process different independent sources corresponding to different input traffic classes. The routers in the network could be identical in their capabilities (homogeneous network) or different (heterogeneous network). Each router may have different capabilities; for example one router could have the ability to correct errors (FEC) and use ARQ, one may use only ARQ but cannot correct errors, a third one may not have MPLS capability.

For the homogeneous case scenario, we found that when the processing factor ( $\tau$ ) and IP factor ( $\xi_1$ ) are high and when MPLS factor ( $\xi_2$ ) is small, IP multicast will perform less efficiently than MPLS in terms of the total packet delay, total delay jitter and the residual packet loss probability. However, when this difference in packet processing time is small IP performs very similar to MPLS. In addition to that when MPLS has higher arrival rate due to MPLS trees establishment control overhead and when the processing factor is small, IP would perform better than MPLS. We have found that the previous results are true for both homogeneous and heterogeneous IP or MPLS multicast networks.

Analysis results revealed that there is a noticeable improvement in QoS defined as the total packet delay, total delay jitter and the residual packet loss probability for a higher priority traffic when MPLS multicasting replaces IP multicasting especially if MPLS factor is small and processing factor is large. However, the difference between the two QoS provided by MPLS and IP becomes minimal for low priority traffic.

In addition to that, we have found that the no FEC or ARQ mechanism is the best mechanism in terms of total packet delay for all IP and MPLS sources, and the hybrid FEC/ARQ unicast mechanism is the best mechanism in terms of the residual packet loss probability for all IP and MPLS sources.

In addition, the residual packet loss probabilities in a complete binary IP and MPLS multicast trees, which consist of  $N$  routers, were evaluated for worst-case scenario, when Automatic Repeat Request (ARQ) with multicast repairs is employed and when DiffServ is adopted. In addition to that, we have derived and compared two other mathematical expressions, which can be used to calculate the residual packet loss probability in IP and MPLS multicast trees. These expressions are the approximate residual loss probability and the exact residual loss probability. Results show that the approximate residual loss probability is very close and represents a good approximation to the exact value over different ranges of intrinsic arrival probabilities and  $N$  values. Hence, this approximate value can be used to calculate the residual loss probability in case of IP or MPLS binary multicast trees, which will lead to less computational efforts.

For the first heterogeneous case scenario, the thesis tries to put an image for the expected future Internet and analyze the performance of this Internet when most of the current IP routers are replaced with MPLS routers. This would create a heterogeneous MPLS network. We compare the performance of this heterogeneous MPLS network with homogeneous IP and homogeneous MPLS networks using analysis tools. The results are general and are evaluated for a wide range of traffic values, priorities, ....etc for a complete binary tree. In addition to that, a Fair Share Policy (FSP), is proposed to ensure proper QoS. Also, Differentiated Services and reliable multicasting are used in this

comparison. We have found that the no FEC or ARQ mechanism is the best mechanism in terms of total packet delay for all IP and MPLS sources, and the hybrid FEC/ARQ unicast mechanism is the best mechanism in terms of the residual packet loss probability for all IP homogeneous, MPLS homogeneous and heterogeneous sources. Also, we have found that when MPLS multicast networks are mixed with some IP routers (heterogeneous network), it will perform less efficiently than homogeneous MPLS multicast networks. In addition to that, when the IP factor is small, the heterogeneous multicast network will perform less efficiently than IP homogeneous networks.

In the second heterogeneous case scenario, a performance comparison between 4 types of multicast networks: homogeneous IP, homogenous MPLS, heterogeneous IP and heterogeneous MPLS is carried using analysis tools. In addition to that a Fair Share Policy (FSP), is proposed to ensure proper QoS. Also, Differentiated Services and reliable FEC/ARQ with unicast repairs mechanism are used in this comparison. We have found that when multicast networks are mixed with some DR routers (heterogeneous network), they will perform more efficiently than homogeneous multicast networks in terms of total packet delay and residual packet loss probability; and this is valid for both IP and MPLS multicast networks. This thesis suggests that as the number of DR routers in the multicast network increases, the network will have better performance in terms of total packet delay and residual packet loss probability.

In our work, we found that simulation results have the same tendency as the analysis results. In addition to that, simulation results were very close to analysis results.

Though in our analysis, we take the complete binary tree case, there is no loss of generality in taking this full binary tree. The analysis is straightforwardly applicable to non-binary and other types of trees. All one has to do is to replace equations (loss) with non-binary trees counterparts.

The tradeoffs between IP and MPLS multicast networks in the different cases under different traffics are clear from the obtained results.

## **8.2 Suggested Future Work**

The design and management of a reliable multicast network is a fundamental key to the success of the QoS provisioning and it includes several open research issues. Many problems need to be solved such as:

- 1- Different types of traffic arrival other than Poisson could be considered; for example Geometric distribution, which could represent file arrival or on/off process which could represent voice or video arrival.
- 2- Other types of queuing could be considered other than FIFO within the same queue and priority queues between different queues which are assumed in our thesis; for example weighted round robin (WRR) or weight fair queuing (WFQ) discipline.
- 3- Other types of heterogeneous networks could be studied; for example, when not all routers have the DiffServ capability.
- 4- A complete heterogeneous multicast network could be studied, where all possible combination of different router types could coexist together.

- 5- In our thesis, a wired network is used where MPLS multicast and IP multicast are carried. However, MPLS multicast in mobile (or wireless) networks could also be considered.
- 6- In our thesis, we consider binary trees only as multicast trees for both IP and MPLS. One may consider using mesh type trees for both IP and MPLS multicast.
- 7- In our thesis, we did not consider routing problems. However, one may consider how to integrate MPLS multicast with current routing protocols such as OSPF, DVMRP or PIM.
- 8- Other problems such as LSP dimensioning, set-up/tear-down procedures could also be studied.

## REFERENCES

- [1] R. Wittmann and M. Zitterbart, " Multicast Communication: Protocols, Programming and Applications ", Morgan Kaufmann Publishers, May 2000.
- [2] T. Maufer and C. Semeria, " Introduction to IP Multicast Routing ", IETF draft, <draft-ietf-mboned-intro-multicast-01.txt>, March 1997.
- [3] B. Quinn et al., " IP Multicast Applications: Challenges and Solutions ", RFC 3170, September 2001.
- [4] S. Paul, " Multicasting on the Internet and Its Applications ", Kluwer Academic Publishers, 1998.
- [5] N. Mir, " A Survey of Data Multicast Techniques, Architectures, and Algorithms ", IEEE Communications Magazine, September 2001.
- [6] Nortel White Papers, " Exploiting Internetwork Multicast Services ", Nortel Networks, 1999.
- [7] S. Deering, " Host Extensions for IP Multicasting ", RFC 1112, August 1989.
- [8] J. Reynold and J. Postel, " Assigned Numbers ", RFC 1700, October 1994.
- [9] G. Armitage, " IP Multicasting over ATM Networks ", IEEE Journal on Selected Areas in Communications, Vol. 15, No. 3, April 1997.
- [10] E. Rosen, A. Viswanathan and R. Callon, " Multiprotocol Label Switching Architecture ", RFC 3031, January 2001.
- [11] J. Lawrence, " Designing Multiprotocol Label Switching Network ", IEEE Communications Magazine, July 2001, pp. 134 – 142.
- [12] A. Viswanathan et al., " Evolution of Multiprotocol Label Switching ", IEEE Communications Magazine, May 1998, pp. 165 – 173.

- [13] F. Faucheur, " IETF Multiprotocol Label Switching (MPLS) Architecture ", 1998 1<sup>st</sup> IEEE International Conference, 1998, pp. 6-15.
- [14] G. Armitage, " MPLS: The Magic Behind the Myths ", IEEE Communications Magazine, January 2000.
- [15] U. Black, " MPLS and Label Switching Networks ", Prentice Hall PTR, 2001.
- [16] B. David and Y. Rekhter, "MPLS Technology and Applications", Morgan Kaufmann Publisher, ISBN: 1-55860-656-4
- [17] L. Mathy, C. Edwards and D. Hutchison, " The Internet: A Global Telecommunications Solution ", IEEE Network, July/August 2000, pp. 46-57.
- [18] L. Anderson et al., " LDP Specification ", RFC 3036, January 2001.
- [19] D. Ooms et al., "Overview of IP Multicast in a MPLS Environment", RFC3353, August 2002.
- [20] D. Ooms and W. Livens, " IP Multicast in MPLS Networks ", Proceedings of the IEEE Conference on High Performance Switching and Routing, June 2000, pp. 301 – 305.
- [21] A. Acharya, F. Griffoul and F. Ansari, " IP Multicast Support in MPLS ", IEEE proceedings, 1999, pp. 211 – 218.
- [22] Z. Zhang et al., " The New Mechanism for MPLS Supporting IP Multicast ", IEEE APCCAS 2000, 2000, pp. 247- 250.
- [23] D. Ooms et al., "MPLS Multicast Traffic Engineering", draft-ooms-mpls-multicast-te-01.txt, February 2002.
- [24] R. Pulley and P. Christensen, " A Comparison of MPLS Traffic Engineering Initiatives ", NetPlane Systems, Inc. Press, 2000.



- [25] K. Almeroth, " The Evolution of Multicast: From the MBone to Interdomain Multicast to Internet2 Deployment ", IEEE Network, January/February 2000.
- [26] C. Diot et al., " Deployment Issues for the IP Multicast Service and Architecture ", IEEE Network, January/February 2000.
- [27] A. Striegel and G. Manimaran, "A Survey of QoS Multicasting Issues ", IEEE Communications Magazine, June 2002.
- [28] K. Obraczka, " Multicast Transport Protocols: A Survey and Taxonomy ", IEEE Communications Magazine, January 1998.
- [29] B. Whetten and G. Taskale, " An Overview of Reliable Multicast Transport Protocol II ", IEEE Network, January/February 2000.
- [30] L. Sahasrabudde and B. Mukherjee, " Multicast Routing Algorithms and Protocols: A Tutorial ", IEEE Network, January/February 2000, pp. 90 - 102.
- [31] A. Agarwal, " Quality of Service (QoS) in the New Public Network Architecture ", IEEE Canadian Review, September 2000.
- [32] D. Wan, " QoS in Next Generation Internet ", The Fourth International Conference/Exhibition on High Performance Computing in the Asia-Pacific Region , Vol. 1, 14-17, May 2000, pp. 65 -70
- [33] X. Xiao and L. Ni, " Internet QoS : A Big Picture ", IEEE Network, pp. 8-18 March/April, 1999.
- [34] R. Braden., D. Clark., and S. Shenker, " Integrated Services in the Internet Architecture: an Overview", RFC 1633, June 1994, <http://www.ietf.org/rfc/rfc1633.txt>
- [35] J. Wrocklawski, " The Use of RSVP with IETF Integrated Services ", RFC 2210, September 1997; <http://www.ietf.org/rfc/rfc2210.txt>

- [36] R. Braden et al., " Resource ReSerVation Protocol (RSVP) -Version 1 Functional Specification ", RFC2205, September 1997, <http://www.ietf.org/rfc/rfc2205.txt>
- [37] J. Wroclawski, " Specification of the Controlled-Load Network Element Service ", RFC 2212, September 1997, , <http://www.ietf.org/rfc/rfc2211.txt>
- [38] S. Shenker, C. Partridge and R. Guerin, " Specification of Guaranteed Quality of Service ", RFC 2212, September 1997, , <http://www.ietf.org/rfc/rfc2212.txt>
- [39] M. Reisslein et al., " A Framework for Guaranteeing Statistical QoS ", IEEE/ACM Transactions on Networking, Vol. 10, No. 1, February 2002.
- [40] A. Danthine, " How to provide QoS in the Next Generation Internet ? ", Proc. Of the International Conference on Communication Technology, WCC-ICCT 2000, Vol. 1, 2000.
- [41] P. Ferguson and G. Huston, "Quality of Service", John Wiley & Sons, 1998.
- [42] S. Blake et al., " An Architecture for Differentiated Services ", RFC 2475, December 1998; <http://www.ietf.org/rfc/rfc2475.txt>
- [43] Y. Bernet et al., " A Framework for Differentiated Services ", Internet draft, <draft-ietf-DiffServ-framework-00.txt>, May 1998.
- [44] K. Nicholas et al., " Definition of the Differentiated Services Field (DS Field)in the IPv4 and IPv6 Headers ", RFC2474, December 1998; <http://www.ietf.org/rfc/rfc2474.txt>
- [45] K. Nichols et al., "Differentiated Services Operational Model and Definitions", Internet draft <draft-nichols-dsopdef-00.txt>, February 1998
- [46] D. Clark and J. Wroclawski, "An Approach to Service Allocation in the Internet", Internet draft <draft-clark-different-svc-alloc-00.txt>, July 1997

- [47] J. Heinanen et al., " Assured Forwarding PHB Group", RFC2597, June 1999;  
<http://www.ietf.org/rfc/rfc2597.txt>
- [48] V. Jacobson et al., " An Expedited Forwarding PHB ", RFC2598, June 1999;  
<http://www.ietf.org/rfc/rfc2598.txt>
- [49] M. Handley et al., " The Reliable Multicast Design Space for Bulk Data Transfer",  
RFC 2887, August 2000, <http://www.ietf.org/rfc/rfc2887.txt>
- [50] M. Luby et al., " The Forward Error Correction (FEC) Building Block", RFC 3452,  
December 2002, <http://www.ietf.org/rfc/rfc3452.txt>
- [51] M. Luby et al., " The Use of Forward Error Correction (FEC) in Reliable Multicast",  
RFC 3453, December 2002, <http://www.ietf.org/rfc/rfc3453.txt>
- [52] S. Lin and D. Costello, " Error Control Coding: Fundamentals and Applications ",  
Prentice-Hall, Inc., 1983.
- [53] W. Shay, " Understanding Data Communications and Networks ", Second edition,  
An International Thomson Publishing Company, 1994.
- [54] T. Lestayo, M. Fernandez and C. Lopez, " Adaptive approach for FEC reliable  
multicast ", Electronics Letters, Vol. 37, No. 22, 25 October 2001, pp. 1333 –1335.
- [55] B. Li, " Reliable multicast transmissions using forward error correction and  
automatic retransmission requests ", 2001 Canadian Conference on Electrical and  
Computer Engineering, Vol. 2, 2001, pp 1145 -1150.
- [56] S. Floyd et al., " A Reliable Multicast Framework for Light-Weight Sessions and  
Application Level Framing, IEEE/ACM Trans. on Networking, 1997.

- [57] S. Kasera, J. Kurose and D. Towsley, " A Comparison of Server-Based and Receiver-Based Local Recovery Approaches for Scalable Reliable Multicast, IEEE Infocom98, 1998.
- [58] S. Kasera, J. Kurose and D. Towsley, " Buffer Requirements and Replacement Policies for Multicast Repair Service ", Proc. of NGC 2000 on Networked Group Communication, USA, 2000, pp. 5-14.
- [59] J. Nonnenmacher, and E. Biersack, " Reliable Multicast: Where to use FEC ", in Proc. IFIP 5<sup>th</sup> Int. Workshop on Protocols for High Speed Networks (PfHSN'96), France, October 1996, pp. 134-148.
- [60] D. Rubenstein, J. Kurose and D. Towsley, " Real-Time Reliable Multicast Using Proactive Forward Error Correction ", Proc. NOSSDAV 98, UK, 1998.
- [61] S. Alwakeel, N. Rikli and A. Alwehaibi, " Evaluation of Fairness Strategies for ATM Congestion Control Policing Mechanisms ", IASTED AMS'99, Carins, Australia, 1999.
- [62] Integral Access White Paper, " MPLS and Next Generation Access Networks ", Integral Access Inc., July 2001.
- [63] S. Akhtar, " Congestion Control in a Fast Packet Switching Network ", Master's thesis, Washington University, 1987.
- [64] E. Rathgeb, " Modeling and Performance Comparison of Policing Mechanisms for ATM Networks ", IEEE Journal on Selected Areas in Communications, Vol. 9, No. 3, April 1991, pp. 325 -334.
- [65] Y. Zheng, et al., "Evolutionary Marking Algorithm: Improving Robustness and Responsiveness of Congestion Control", IEICE Trans. Communications, Vol. E86-B. No. 2, February 2003.

- [66] S. Ueno, T. Kato and K. Suzuki, " Analysis of Internet Multicast Traffic Performance Considering Multicast Routing Protocol ", Proc. of the International Conference on Network Protocols, November, 2000.
- [67] M. Yamamoto, J. Kurose, D. Towsley and H. Ikeda, " A Delay Analysis of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols ", IEEE INFOCOM, Vol. 2, April 1997.
- [68] E. Altman et al., " Queueing analysis of simple FEC schemes for IP Telephony ", IEEE INFOCOM, 2001.
- [69] D. Rubenstein, S. Kasper, D. Towsley and J. Kurose, " Improving Reliable Multicast Using Active Parity Encoding Services (APES) ", INFOCOM'99, USA, April, 1999.
- [70] D. Rubenstein, S. Kasper, D. Towsley and J. Kurose, " Improving Reliable Multicast Using Active Parity Encoding Services (APES) ", Technical Report 98-79, Department of Computer Science, University of Massachusetts, July 1998.
- [71] T. Noguchi, and M. Yamamoto, " Reliable Multicast Protocol Applying Local FEC ", IEICE Trans. Communications., Vol. E86-B. No. 2, February 2003.
- [72] S. Kasper, G. Hjalmtysson, D. Towsley, and J. Kurose, " Scalable Reliable Multicast Using Multiple Multicast Channels ", IEEE/ACM TON, Vol. 8, No. 3, June 2000.
- [73] R. Bless and K. Wehrle, " IP Multicast in Differentiated Services Networks ", draft-[bless-diffserv-multicast-06.txt](#), February 2003.
- [74] R. Bless and K. Wehrle, " Group Communication in Differentiated Services Networks ", Proceedings of First IEEE/ACM International Symposium on Cluster Computing and the Grid, pp. 618 -625, May 2001.

- [75] A. Striegel and G. Manimaran, " A scalable approach for DiffServ multicasting", ICC 2001, IEEE International Conference on Communications, Vol. 8, pp 2331, 2001.
- [76] A. Striegel and G. Manimaran, " A Scalable Protocol for Member Join/Leave in DiffServ Multicast ", Proc. of Local Computer Networks (LCN), Tampa, Florida, November 2001
- [77] R. Bless and K. Wehrle, " DS Multicast Router Extension ", draft-bleess-diffserv-mcast-routerext-00.txt, July 2001
- [78] H. Su and R. Hwang, " Multicast provision in a differentiated services network ", 15th International Conference on Information Networking, pp. 189-196, 2001.
- [79] G. Biachi et al., " QUASIMODO: Quality of Service-Aware Multicasting over DiffServ and Overlay Networks ", IEEE Network, January/February 2003.
- [80] L. Zhi and P. Mohapatra, " QoS-aware Multicasting in DiffServ Domains ", IEEE GLOBECOM'02, Vol. 3, No. 17-21, pp. 2118-2122, 2002.
- [81] B. Yang and P. Mohapatra, " Multicasting in Differentiated Services Domains ", IEEE GLOBECOM, 2002.
- [82] M. Gupta and M. Ammar, " Providing Multicast Communications in a Differentiated Services Network Using Limited Branching Techniques ", Tech. Report GIT-CC-02-27, May 2002.
- [83] J. Cui, A. Fei, M. Gerla and M. Faloutsos " Aggregated Multicast: A Scheme to Reduce Multicast States ", draft-cui-multicast-aggregation-01.txt, September 2002.
- [84] A. Fei, J. Cui , M. Gerla, and M. Faloutsos" Aggregated Multicast: an Approach to Reduce Multicast State ", Proceedings of the Sixth Global Internet Symposium, GI'01, November 2001.

- [85] J. Chung et al., " MPLS Multicasting Through Enhanced LDP and RSVP-TE Control ", The 45<sup>th</sup> Midwest Symposium on Circuits and Systems, MWSCAS02, Vol. 3, pp. 93-96, 2002.
- [86] B. Yang and P. Mohapatra, " Edge Router Multicasting with MPLS Traffic Engineering ", 10<sup>th</sup> IEEE International Conference on Networks, ICON02, August 2002.
- [87] A. Boudani and B. Cousin, " A new Approach to construct Multicast Trees in MPLS Networks ", 7<sup>th</sup> International Symposium on Computers and Communications, ISCC02, July 2002.
- [88] A. Boudani et al., "Multicast Routing Simulator over MPLS Networks ", 36<sup>th</sup> Annual Simulation Symposium, April 2003.
- [89] Y. Oh et al., " Scalable MPLS Multicast Using Label Aggregation in Internet Broadcasting System ", 10<sup>th</sup> International Conference on Telecommunications, ICT03, Vol. 1, March 2003.
- [90] D. Cheng, " RSVP-TE: Extensions to RSVP for Multicast LSP Tunnels ", draft-cheng-mpls-rsvp-multicast-er-00.txt, October 2001.
- [91] J. Chung, " RSVP-TE Extensions for MPLS Multicasting Services ", draft-chung-mpls-rsvp-multicasting-00.txt, February 2002.
- [92] F. Faucheur, et al, " Multi-Protocol Label Switching (MPLS) Support of Differentiated Services ", RFC3270, May 2002
- [93] J. Cui, J. Kim, A. Fei, M. Faloutsos and M. Gerla, " Scalable QoS Multicast Provisioning in Diff-Serv- Supported MPLS Networks ", IEEE GLOBECOM, Vol. 2, November 2002.

- [94] N. Rouhana and E. Horlait, " Differentiated services and integrated services use of MPLS ", ISCC 2000. Fifth IEEE Symposium on Computers and Communications, pp. 194-199, 2000.
- [95] I. Andrikopoulos and G. Pavolou, " Supporting differentiated services in MPLS networks ", IWQoS'99, Seventh International Workshop on Quality of Service, pp. 207-215, 1999.
- [96] M. Moh, B. Wei and J. Zhu, " Supporting differentiated services with per-class traffic engineering in MPLS ", Tenth International Conference on Computer Communications and Networks, pp. 354-360, 2001.
- [97] Z. Jing, L. Li and H. Sun, " Supporting Differentiated Services in MPLS-based ATM Switches ", APCC/OECC'99, Fifth Asia-Pacific Conference on Communications and Fourth Optoelectronics and Communications Conferences, pp.91-93 Vol.1, 1999.
- [98] K. Wang and A. Agarwal, " Multicast Traffic Merging in DiffServ-Supported MPLS Networks ", IEEE CCECE'03, May 2003.
- [99] R. Law and S. Raghavan, " DiffServ and MPLS- Concepts and Simulation ", Final Report Submitted to Virginia Tech, 2001.
- [100] E. Dinan, D. Awduche and B. Jabbari, " Analytical Framework for Dynamic Traffic Partitioning in MPLS Networks ", IEEE International Conference on Communications, Vol. 3, June 2000.
- [101] J. Manner, et al., " Provision of QoS in heterogeneous wireless IP access network", Personal, Indoor and Mobile Radio Communications. The 13<sup>th</sup> IEEE International Symposium on, Vol. 2, 15-18, pp. 530-534, September 2002.



- [102] T. Janevski and B. Spasenovski, " QoS analyses of multimedia traffic in heterogeneous wireless IP networks ", IEEE International Conference on Personal Wireless Communications, pp. 207-211, 17-20 December 2002.
- [103] Y. Gwon, et al., " Adaptive approach for locally optimized IP handoffs across heterogeneous wireless networks ", 4<sup>th</sup> International Workshop on Mobil and Wireless Communications Network, pp. 475-479, 9-11 September 2002.
- [104] G. Xia, et al., " End-to-end QoS provisioning in mobile heterogeneous networks", IEEE Wireless Communications and Networking, WCNC 2003., Vol. 2, pp. 1361-1366, 16-20 March 2003.
- [105] D. Adami, et al., " A Quality of Service Guarantee in IP Satellite Environment: Experimental Experience in the CNIT-ASI Project ", Integration of multimedia services on heterogeneous satellite networks, Global Telecommunications Conference, Glob 2000.
- [106] M. Alam, R. Prasad and J Farserotu, " Quality of service among IP-based heterogeneous networks ", IEEE Personal Communications , Vol. 8, No. 6 , pp. 18-24, December 2001.
- [107] Y. Breitbart, et al., " Topology discovery in heterogeneous IP networks ", 19<sup>th</sup> Proceedings Annual Joint Conference of the IEEE Computer and Communications Societies., Vol. 1, pp. 265-274, 26-30, March 2000.
- [108] J. Schmitt et al., " Heterogeneous multicast in heterogeneous QoS networks ", 2001 Proceedings. Ninth IEEE International Conference on Networks, pp. 349 –354, 2001.

- [109] M. Bech, et al., " An Exposed Approach to Reliable Multicast in Heterogeneous Logistical Networks ", Proceeding of 3<sup>rd</sup> IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGrid 2003. pp. 526-533, 2003.
- [110] M. Cha, et al., " A Multicasting Strategy for Multiple Heterogeneous Networks ", Proceedings of 1997 International Conference on Information, Communications and Signal processing, ICICS 97, September 1997.
- [111] C. Yatin, et al., " RMX: Reliable Multicast for Heterogeneous Networks ", IEEE INFOCOM 2000.
- [112] Y. Pei and J. Modestion, " Use of Concatenated FEC Coding for Real-Time Packet Video over Heterogeneous Wired-to-Wireless IP networks ", Proceedings of the 2003 International Symposium on Circuits and Systems. ISCAS '03, Vol. 2, May 25-28, 2003.
- [113] A. AlWehaibi, A. Agarwal, M. Kadoch and A. ElHakeem, " Accommodations of QoS DiffServ Over IP and MPLS Networks ", WSEAS Trans. on Systems, Vol. 2, Issue 2, April 2003.
- [114] A. Alwehaibi, M. Kadoch and A. ElHakeem, " Average Multicast Delay for FEC/DiffServ Over IP and MPLS homogeneous Networks ", Proc. of the 10<sup>th</sup> IEEE International Conference on Electronics, Circuits and Systems, ICECS, December 2003, UAE.
- [115] A. Alwehaibi, A. Agarwal, S. AlWakeel, N. Rikli and A. ElHakeem, " Performance Behavior Evaluation of Internet Congestion Control Policing Mechanisms ", IEEE Canadian Conference on Electrical and Computer Engineering, CCECE, May 2003, Canada.

- [116] T. Saadawi, M. Ammar and A. Elhakeem, " Fundamentals of Telecommunication Networks ", New York, NY: John Wiley & Sons Inc, 1994
- [117] L. Kleinrock, " Queuing Systems ", Volume I and II, John Wiley, New York, NY, 1975.
- [118] G. Bolch, S. Greiner, H. Meer and K. Trivedi, " Queueing Networks and Markov Chains ", John Wiley and Sons, 1998.
- [119] J. Daigle, "Queueing Theory for Telecommunications", Addison-Wesley Publishing Company Inc., 1992.
- [120] R. Wolff, " Stochastic Modeling and The Theory of Queues ", Prentice-Hall Inc., 1989.
- [121] N. Jaiswal, " Priority Queues ", Academic Press, 1968.
- [122] A. Garcia, " Probability and Random Processes for Electrical Engineering ", Second Edition, Addison-Wesley Publishing Company Inc., 1994.
- [123] R. Walpole, R. Myers and S. Myers, " Probability and Statistics for Engineers and Scientists ", Sixth Edition, Prentice-Hall Inc., 1998.
- [124] D. Jordan and P. Smith, " Mathematical Techniques ", Oxford University Press, 1994.
- [125] S. Wicker, " Error Control Strategies- for Digital Communication and Storage", Prentice Hall, 1995.
- [126] A. Alwehaibi, M. Kadoch and A. ElHakeem, "Computation of the Residual Packet Loss Probability in a Binary Multicast Tree ", IEEE Canadian Conference on Electrical and Computer Engineering, CCECE, May 2003, Canada.

- [127] A. Alwehaibi, M. Kadoch, A. Agarwal and A. ElHakeem, " Residual Packet Loss Probability for Diffserv over IP and MPLS Multicast Trees", In Press, International Journal of Computer Research, November 2003.
- [128] G. Gordon, " System Simulation ", Second Edition, Prentice Hall Inc., 1978.
- [129] A. Law and W. Kelton, " Simulation Modeling and Analysis ", Second Edition, McGraw Hill, 1991.