Active Services QoS Support for Multimedia Communications

Sertac Eminsoy

Submitted for the Degree of Doctor of Philosophy from the University of Surrey

Centre for Communication Systems Research School of Electronics and Physical Sciences University of Surrey Guildford, Surrey GU2 7XH, UK

June 2005

© Sertac Eminsoy 2005
Abstract

Present-day Internet Protocol (IP) networks employ Quality of Service (QoS) architectures that can mainly provide network-level support for multimedia communications. Although these approaches are effective in delivering both soft and hard service guarantees, they are usually inefficient in satisfying specific user and application-level QoS requirements. This is due to the fact that such networks are designed with the principle of implementing the main intelligence and media processing at the end-points. Moreover, introduction of new services depend on cumbersome standardisation process which hinders their timely deployment. Furthermore, increasing heterogeneity of networks and end systems makes it difficult to provide end-to-end QoS guarantees. Therefore, providing advanced support for user applications require the networks to evolve into a more flexible architecture which can shift some of the intelligence and processing load away from the end-user terminals. It is envisaged that this evolution will take place in line with the active networking approaches where distributed processing of multimedia applications will be performed inside the networks through the use of dynamically deployable content adaptation services.

Looking from a user-centric point-of-view, this thesis proposes to enhance the perceived quality of video-based applications by utilising a series of visual content adaptation services that can be deployed in the active network nodes. These services are based on a real-time video transcoding system that is capable of applying certain rate-control and error-resilience operations on the input video streams. This enables video transmission over heterogeneous networks according to their respective bandwidth and error-resilience requirements. In addition, rate-control functionality of the transcoder is presented as an alternative congestion control mechanism to the traditional traffic conditioning approaches. Furthermore, an adaptive error resilience scheme is presented, which is responsive to the detected video scene activity and the reported channel error conditions of the wireless network. Extensive computer simulations demonstrate the effectiveness of the visual content adaptation services in terms of improving the perceived QoS of multimedia applications over both wired and wireless networks.

Key words: Multimedia Communications, QoS, Active Networks, Differentiated Services, Active Services.

Email: s.eminsoy@surrey.ac.uk
www: http://www.ee.surrey.ac.uk/Ilab
Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Ahmet M. Kondoz whose guidance and support were invaluable. He has been a mentor to me, and his dedication to his work and family will always be my source of inspiration to achieve the best in the life.

I am very much indebted to my colleague and friend Dr. Şafak Doğan for his contributions towards my work. I would also like to thank all my colleges in the I-lab who helped and supported me throughout my research, and provided a friendly working environment.

I am most grateful to my beloved family without whose love, care, and sacrifice I would not be here and to whom I owe everything. Lastly, I would like to take this opportunity to express my deepest appreciation to my girlfriend Catherine for her love, patience, and encouragement.
Contents

Summary ........................................................ i
Acknowledgements ........................................................... ii
Contents ........................................................................................................ iii
List of Figures ............................................................................................... viii
List of Tables ............................................................................................... xi
List of Publications ....................................................................................... xii
List of Abbreviations .................................................................................. xiii

Chapter 1 Introduction

1.1 Background and Objectives ........................................................................ 1
1.2 Quality Assessment by Performance Evaluation .................................... 2
1.3 Source Material ....................................................................................... 3
1.4 Original Achievements and Contributions ............................................. 5
1.5 Thesis Outline ......................................................................................... 6

Chapter 2 Multimedia Communications

2.1 Overview ............................................................................................... 8
2.2 Quality of Service ................................................................................ 8
  2.2.1 Network QoS Parameters ................................................................. 11
  2.2.2 The User Perspective ....................................................................... 13
  2.2.3 Application QoS Metrics .................................................................. 14
2.3 Classification and Requirements of Multimedia Applications ............ 15
  2.3.1 Presentational Characteristics ............................................................ 17
  2.3.2 Networking Characteristics ............................................................... 18
  2.3.3 Voice-Based Applications ................................................................. 19
    2.3.3.1 QoS Requirements for VoIP ......................................................... 19
  2.3.4 Video-Based Applications ............................................................... 21
    2.3.4.1 QoS Requirements for Video-Based Applications .................. 22
    2.3.4.2 Effects of Packet Loss and Bit-Errors on the Video Quality ...... 24
    2.3.4.3 ISO MPEG-4 ............................................................................ 25
2.4 Multimedia Transport over Wired and Wireless Networks .................. 27
  2.4.1 Communication Protocols ................................................................. 28
    2.4.1.1 Internet Protocol ................................................................. 28
## Contents

2.4.1.2 User Datagram Protocol ................................................................. 28  
2.4.1.3 Real-Time Transport Protocol ......................................................... 28  
2.4.1.4 Transport Control Protocol ............................................................... 29  
2.4.2 QoS Architectures for IP-Networks ....................................................... 29  
2.4.3 QoS for Wireless Multimedia ............................................................... 31  
2.5 QoS Adaptation Techniques ................................................................. 33  
2.5.1 End-User Based Adaptation ............................................................... 34  
2.5.2 Network Node Based Adaptation ........................................................ 35  
2.6 Evolution of Networks and Applications ............................................... 37  
2.7 Concluding Remarks and Recommendations ........................................ 40  

### Chapter 3 Active Networks

3.1 Overview .................................................................................................. 42  
3.2 Background ............................................................................................... 42  
3.2.1 What is Active Networking? ............................................................... 42  
3.2.2 Advantages of Active Networks ........................................................ 45  
3.2.3 Active Networking and end-to-End Arguments .................................. 47  
3.3 Active Network Architecture ................................................................. 49  
3.3.1 Design Objectives ................................................................................ 49  
3.3.2 Active Nodes ........................................................................................ 49  
3.3.3 Programming Models ......................................................................... 51  
3.3.3.1 Code Distribution Mechanisms ....................................................... 52  
3.3.3.2 Code Encoding ................................................................................. 54  
3.4 Active Processing ..................................................................................... 55  
3.4.1 Cost of Active Processing ................................................................. 56  
3.4.2 Granularity of Control ....................................................................... 57  
3.5 Deployment Strategies ............................................................................ 58  
3.5.1 How to Deploy ..................................................................................... 58  
3.5.2 Where to Deploy ................................................................................ 59  
3.6 Applications of Active Networks ........................................................... 60  
3.6.1 QoS .................................................................................................... 60  
3.6.1.1 Routing ........................................................................................... 61  
3.6.1.2 Traffic Conditioning ....................................................................... 61  
3.6.1.3 Congestion Control ........................................................................ 62  
3.6.1.4 Delay and Jitter Control ................................................................. 62  
3.6.1.5 Error Control ................................................................................ 63  
3.6.1.6 Caching and Retransmission ......................................................... 63  

iv
3.6.1.7 Personalisation and Context Awareness ........................................... 64
3.6.2 Protocol Deployment and Signalling ............................................................. 65
3.6.3 Network Management .................................................................................... 65
3.7 Technical and Practical Challenges ....................................................................... 66
3.7.1 Technical Challenges ..................................................................................... 66
3.7.2 Practical Challenges ....................................................................................... 67
3.8 Concluding Remarks and Recommendations ........................................................ 68

Chapter 4 Differentiated Services 71
4.1 Overview ................................................................................................................ 71
4.2 Conceptual Overview............................................................................................. 72
4.2.1 Motivation...................................................................................................... 72
4.2.2 Service Model ................................................................................................ 72
4.2.3 Scalability ...................................................................................................... 75
4.2.4 Packet Differentiation .................................................................................... 75
4.2.5 Providing End-to-End Services...................................................................... 76
4.3 Differentiated Services Architectural Model ......................................................... 77
4.3.1 Traffic Classification and Conditioning...........................................  78
  4.3.1.1 Classifier ............................................................................................. 79
  4.3.1.2 Meter .............................................................................................. 80
  4.3.1.3 Marker................................................................................................. 80
  4.3.1.4 Shaper.................................................................................................. 80
  4.3.1.5 Dropper/Policer................................................................................... 80
4.3.2 Forwarding Behaviours.................................................................................. 81
  4.3.2.1 Default Behaviour ......................................................................... 82
  4.3.2.2 Class Selector Behaviours ................................................................. 82
  4.3.2.3 Expedited Forwarding......................................................................... 82
  4.3.2.4 Assured Forwarding............................. 83
  4.3.2.5 Lower Effort Forwarding .................................................................... 84
4.4 Traffic Control Mechanisms.................................................................................. 85
  4.4.1 Traffic Conditioning Algorithms ................................................................. 85
    4.4.2.1 Token and Leaky Bucket Algorithms ................................................. 86
    4.4.2.1 Absolute and Algorithmic Droppers................................................... 99
4.4.2 Queuing and Scheduling .............................................................................. 90
  4.4.2.1 Active Queue Management................................................................. 91
  4.4.2.2 Queue Scheduling Disciplines ............................................................ 95
4.4.3 A Practical Implementation ............................................................................. 101
### Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.4.4 Routing and MPLS</td>
<td>102</td>
</tr>
<tr>
<td>4.5 Traffic Categories and Service Classes</td>
<td>104</td>
</tr>
<tr>
<td>4.5.1 Telephony</td>
<td>106</td>
</tr>
<tr>
<td>4.5.2 Multimedia Conferencing</td>
<td>106</td>
</tr>
<tr>
<td>4.5.3 Multimedia Streaming</td>
<td>107</td>
</tr>
<tr>
<td>4.5.4 Low Latency Data</td>
<td>107</td>
</tr>
<tr>
<td>4.5.5 High Throughput Data</td>
<td>108</td>
</tr>
<tr>
<td>4.5.6 Standard Service Class</td>
<td>108</td>
</tr>
<tr>
<td>4.5.7 Low Priority Data</td>
<td>108</td>
</tr>
<tr>
<td>4.5.8 Summary</td>
<td>108</td>
</tr>
<tr>
<td>4.6 Differentiated Services QoS Support in UMTS</td>
<td>110</td>
</tr>
<tr>
<td>4.7 Differentiated Services for Multimedia</td>
<td>113</td>
</tr>
<tr>
<td>4.7.1 Granularity of Aggregation</td>
<td>114</td>
</tr>
<tr>
<td>4.7.2 Prioritisation Based on Packet Contents</td>
<td>115</td>
</tr>
<tr>
<td>4.7.3 Prioritisation Based on Packet Categories</td>
<td>117</td>
</tr>
<tr>
<td>4.7.4 Discussion</td>
<td>118</td>
</tr>
<tr>
<td>4.8 A Performance Analysis of Differentiated Services for Multimedia</td>
<td>119</td>
</tr>
<tr>
<td>4.8.1 The Simulation Model</td>
<td>119</td>
</tr>
<tr>
<td>4.8.1.1 Modelling the Background Traffic</td>
<td>121</td>
</tr>
<tr>
<td>4.8.2 Experimental Procedure</td>
<td>122</td>
</tr>
<tr>
<td>4.8.3 Results and Discussion</td>
<td>123</td>
</tr>
<tr>
<td>4.9 Concluding Remarks and Recommendations</td>
<td>128</td>
</tr>
</tbody>
</table>

### Chapter 5 Active Services QoS Support

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1 Overview</td>
<td>131</td>
</tr>
<tr>
<td>5.2 Active Services</td>
<td>131</td>
</tr>
<tr>
<td>5.3 Transcoding Class Services</td>
<td>133</td>
</tr>
<tr>
<td>5.3.1 Transcoding Architecture</td>
<td>134</td>
</tr>
<tr>
<td>5.3.1.1 Operation of the Transcoder</td>
<td>136</td>
</tr>
<tr>
<td>5.3.1.2 Transcoder Capabilities and its Design Criteria</td>
<td>137</td>
</tr>
<tr>
<td>5.3.2 Rate-Control Service</td>
<td>138</td>
</tr>
<tr>
<td>5.3.2.1 Rate-Control Algorithm</td>
<td>139</td>
</tr>
<tr>
<td>5.3.2.2 Rate-Control Service Performance Tests</td>
<td>142</td>
</tr>
<tr>
<td>5.3.3 Error-Resilience Services</td>
<td>150</td>
</tr>
<tr>
<td>5.3.3.1 Video Packet Resynchronisation</td>
<td>151</td>
</tr>
<tr>
<td>5.3.3.2 Header Extension Code</td>
<td>152</td>
</tr>
<tr>
<td>5.3.3.3 Data Partitioning</td>
<td>153</td>
</tr>
</tbody>
</table>
List of Figures

Chapter 1

Figure 1.1 Images from the Standard Test Sequences: (a) 6th, 114th, 288th, 399th frames of “Foreman”, (b) 6th, 114th, 256th, 399th frames of “Students” ........................................... 4

Figure 1.2 Images from the Non-Standard Test Sequences: (a) 1st, 114th, 225th, 351st frames of “Denmark”, (b) 1st, 165th, 139th, 393rd frames of “Interview” ......................... 5

Chapter 3

Figure 3.1 Operation of Passive Networks ............................................................................... 43
Figure 3.2 Operation of Active Networks .................................................................................. 45
Figure 3.3 Cascaded Active Node Architecture ....................................................................... 51
Figure 3.4 Hybrid Code Distribution in an Active Network Domain ........................................ 53

Chapter 4

Figure 4.1 The DiffServ Architecture ..................................................................................... 78
Figure 4.2 Logical Structure of a Packet Classifier and Traffic Conditioner ............................. 79
Figure 4.3 Effect of Shaping on the Traffic Rate ..................................................................... 81
Figure 4.4 Effect of Policing Operation on the Traffic Rate .................................................... 81
Figure 4.5 The Simple Token Bucket Operation ..................................................................... 86
Figure 4.6 Implementation of Two Rate Three Colour Marker .............................................. 88
Figure 4.7 Algorithmic Dropper Attached to the End of a FIFO Queue ................................. 90
Figure 4.8 A Simple RED Drop Profile ................................................................................ 92
Figure 4.9 Three Level RED Drop Profile for AF Service ..................................................... 94
Figure 4.10 A Three Level PQ ............................................................................................. 96
Figure 4.11 WFQ Scheduling with VFT Calculations ............................................................... 99
Figure 4.12 A Practical DiffServ Edge Node Model ............................................................... 102
Figure 4.13 DiffServ Capable UMTS Network ..................................................................... 111
Figure 4.14 (a) Original Akiyo Sequence, (b) Background VO, (c) Foreground VO ............... 116
Figure 4.15 (a) 20kbps Encoded with no Packet loss, (b) Both Foreground and Background Subject to Average 7.4% Packet Loss, (c) DiffServ Case with no Packet Loss for the Foreground and Average 9.5% Packet Loss for the Background ............................................... 117
Figure 4.16 The Network Topology of the Simulation Model .................................................. 120
Figure 4.17 Operation of the DWRR Scheduler ................................................................. 123
Figure 4.18 Packet Loss Rate of Multimedia Traffic with the Increasing Volume of Background Traffic ...................................................................................................................... 124
Figure 4.19 Average End-to-End Delay Experienced by Multimedia Traffic with the Increasing Volume of Background Traffic ...................................................................................... 125
Figure 4.20 Quality Degradation in the Video Traffic with the Increasing Volume of Background Traffic ........................................................................................................................... 126
Figure 4.21 Quality Degradation in the Voice Traffic with the Increasing Volume of Background Traffic ............................................................................................................................. 127
Figure 4.22 Delay Performance of the Applications Using the DWRR Scheduler ................. 128

Chapter 5

Figure 5.1 Block Diagram of the Cascaded Pixel-Domain MPEG-4 Video Transcoding Architecture ........................................................................................................................................ 136
Figure 5.2 Effects of the RC Service on the Transcoder’s Throughput for the “Foreman” Stream ........................................................................................................................................... 144
Figure 5.3 Effects of the RC Service on the Transcoder’s Throughput for the “Students” Stream ..................................................................................................................................................... 145
Figure 5.4 Service PSNR Results for the “Foreman” Stream at Various Bit-Rates ................. 145
Figure 5.5 Service PSNR Results for the “Students” Stream at Various Bit-Rates ................. 146
Figure 5.6 Subjective Test Results of the 2nd, 36th, 96th, and 133rd Frames of the Transcoded “Foreman” Stream: (a) VBR at avg. 445kbits/s, (b) with RC = 445kbits/s, (c) with RC = 128kbits/s, (d) with RC = 96kbits/s, (e) with RC = 64kbits/s, (f) with RC = 32kbits/s ............................................................................................................. 148
Figure 5.7 Subjective Test Results of the 2nd, 38th, 86th, and 133rd Frames of the Transcoded “Students” Stream: (a) VBR at avg. 230kbits/s, (b) with RC = 230kbits/s, (c) with RC = 128kbits/s, (d) with RC = 96kbits/s, (e) with RC = 64kbits/s, (f) with RC = 32kbits/s ............................................................................................................. 149
Figure 5.8 The Structure of a Video Packet .............................................................................. 151
Figure 5.9 Distribution of Video Packets over a Video Frame ................................................ 152
Figure 5.10 The Structure of a Video Packet in Data Partitioning Mode .................................. 153
Figure 5.11 Standard MPEG-4 AIR Algorithm (1 = motion, 0 = no motion) ......................... 155
Figure 5.12 Experimental Scenario for Error-Resilience Services ........................................ 156
Figure 5.13 Effects of Error-Resilience Services on the “Foreman” Stream.............................. 161
Figure 5.14 Effects of Error-Resilience Services on the “Students” Stream.............................. 161
Figure 5.15 Subjective Test Results of the 30th, 60th, 90th and 120th Frames of the Transcoded “Foreman” Stream for a particular seed at Eb/No = 9dB: (a) error-free, (b) VPR+HEC+DP+10AIR, (c) VPR+HEC+DP, (d) VPR+HEC, (e) no ER services applied ........................................................................................................................................... 163
Figure 5.16 Subjective Test Results of the 30th, 60th, 90th and 120th Frames of the Transcoded “Students” Stream for a particular seed at Eb/No = 9dB:
(a) error-free, (b) VPR+HEC+DP+10AIR,  (c) VPR+HEC+DP,
(d) VPR+HEC, (e) no ER services applied .............................................  166

Figure 5.17 NAI Computation for the “Foreman” Stream .............................................  168
Figure 5.18 NAI Computation for the “Students” Stream .............................................  168
Figure 5.19 Scale Factor and Range Index Analysis for the “Foreman” Sequence ..............  171
Figure 5.20 Scale Factor and Range Index Analysis for the “Students” Sequence ..............  171
Figure 5.21 Number of Intra-Refresh Blocks Computed for the “Foreman” and the “Students” Streams .................................................................  172

Figure 5.22 NAI Computation for the “Denmark” and the “Interview” Streams ..............  174
Figure 5.23 Traffic Conditioning Mechanism Using the Transcoding Class Services ..........  184
Figure 5.24 Traffic Pattern of the Background-Stream ..................................................  188

Figure 5.25 PSNR Plots of With and Without Transcoding Cases for the “Interview” Stream .................................................................................................  188

Figure 5.26 Subjective Test Results of the Experiments using the “Interview” Sequence:
(a) 2nd, 36th, 60th, and 90th frames in the first experiment,
(b) 139th, 159th, 179th, and 228th frames in the first experiment,
(c) 286th, 327th, 366th, and 398th frames in the first experiment,
(d) 2nd, 36th, 60th, and 90th frames in the second experiment,
(e) 139th, 159th, 179th, and 228th frames in the second experiment,
(f) 286th, 327th, 366th, and 398th frames in the second experiment ............................  190
List of Tables

Chapter 2

Table 2.1 Indication of Suitable Performance Targets for Video-Based Applications .......... 24

Chapter 4

Table 4.1 Delay Requirements of Different User Traffic Types ........................................... 105
Table 4.2 Service Classes and Corresponding Application Examples .................................... 106
Table 4.3 Summary of the Service Classes and the QoS Mechanisms Used ......................... 109

Chapter 5

Table 5.1 Average PSNR Values for the RC Service ............................................................ 147
Table 5.2 Average PSNR Values for the ER Services .......................................................... 159
Table 5.3 Test Results for the “Foreman” Stream Using AIR and SC-AIR ............................ 176
Table 5.4 Test Results for the “Students” Stream Using AIR and SC-AIR ............................ 177
Table 5.5 Test Results for the “Interview” Stream Using AIR and SC-AIR ............................ 178
Table 5.6 Test Results for the “Denmark” Stream Using AIR and SC-AIR ............................ 179
Table 5.7 Transcoder’s Computational Complexity Analysis for the “Foreman” Sequence .......... 182
Table 5.8 Transcoder’s Computational Complexity Analysis for the “Students” Sequence .......... 182
List of Publications


<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>3G</td>
<td>third generation</td>
</tr>
<tr>
<td>ACC</td>
<td>active congestion control</td>
</tr>
<tr>
<td>AF</td>
<td>assured forwarding</td>
</tr>
<tr>
<td>AIR</td>
<td>adaptive intra refresh</td>
</tr>
<tr>
<td>AN</td>
<td>active networking</td>
</tr>
<tr>
<td>ANEP</td>
<td>active network encapsulation protocol</td>
</tr>
<tr>
<td>AP</td>
<td>active processing</td>
</tr>
<tr>
<td>AQM</td>
<td>active queue management</td>
</tr>
<tr>
<td>AS</td>
<td>active services</td>
</tr>
<tr>
<td>ATM</td>
<td>asynchronous transfer mode</td>
</tr>
<tr>
<td>B</td>
<td>bi-directional</td>
</tr>
<tr>
<td>BE</td>
<td>best effort</td>
</tr>
<tr>
<td>BER</td>
<td>bit-error-rate</td>
</tr>
<tr>
<td>BG</td>
<td>background</td>
</tr>
<tr>
<td>BS</td>
<td>base station</td>
</tr>
<tr>
<td>C/I</td>
<td>carrier-to-interference ratio</td>
</tr>
<tr>
<td>CBQ</td>
<td>class based queueing</td>
</tr>
<tr>
<td>CBS</td>
<td>committed burst size</td>
</tr>
<tr>
<td>CCSR</td>
<td>Centre for Communication Systems Research</td>
</tr>
<tr>
<td>CDN</td>
<td>content distribution networks</td>
</tr>
<tr>
<td>CIF</td>
<td>common intermediate format</td>
</tr>
<tr>
<td>CIR</td>
<td>committed information rate</td>
</tr>
<tr>
<td>CN</td>
<td>core network</td>
</tr>
<tr>
<td>CSN</td>
<td>content services networks</td>
</tr>
<tr>
<td>DCT</td>
<td>discrete cosine transform</td>
</tr>
<tr>
<td>DiffServ</td>
<td>differentiated services</td>
</tr>
<tr>
<td>DS</td>
<td>differentiated services</td>
</tr>
<tr>
<td>DSCP</td>
<td>differentiated services code point</td>
</tr>
<tr>
<td>DWRR</td>
<td>deficit weighted round robin</td>
</tr>
</tbody>
</table>
EE  execution environment
EBS  excess burst size
EDGE enhanced data rates for global system for mobile telecommunication evolution
EF  expedited forwarding
ER  error-resilience
FEC  forward error correction
FIFO  first in first out
FQ  fair queueing
GGSN gateway GPRS support node
GOP  group of pictures
GPRS general packet radio service
GPS  generalized processor sharing
GSM global system for mobile telecommunications
HVS  human visual system
I  intra
IDCT  inverse discrete cosine transform
IETF  Internet Engineering Task Force
IntServ  integrated services
IP  Internet protocol
ISO  International Standards Organisation
ITU  International Telecommunication Union
LE  lower effort forwarding
LSP  label switched path
LSR  label switched router
MB  macroblock
MEE management execution environment
MOS  mean opinion score
MPEG  Motion Picture Experts Group
MPLS  multi-protocol label switching
MSE  mean-square-error
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>MTU</td>
<td>media transmission unit</td>
</tr>
<tr>
<td>MV</td>
<td>motion vector</td>
</tr>
<tr>
<td>NAPI</td>
<td>network application programming interface</td>
</tr>
<tr>
<td>NodeOS</td>
<td>node operating system</td>
</tr>
<tr>
<td>OSI</td>
<td>open systems interconnection</td>
</tr>
<tr>
<td>P</td>
<td>inter, predictive</td>
</tr>
<tr>
<td>PDB</td>
<td>per-domain behaviour</td>
</tr>
<tr>
<td>PDP</td>
<td>packet data protocol</td>
</tr>
<tr>
<td>PHB</td>
<td>per-hop behaviour</td>
</tr>
<tr>
<td>PIR</td>
<td>peak information rate</td>
</tr>
<tr>
<td>PLR</td>
<td>packet-loss-ratio</td>
</tr>
<tr>
<td>PQ</td>
<td>priority queueing</td>
</tr>
<tr>
<td>PSNR</td>
<td>peak-to-peak signal-to-noise ratio</td>
</tr>
<tr>
<td>QCIF</td>
<td>quarter common intermediate format</td>
</tr>
<tr>
<td>QoS</td>
<td>quality of service</td>
</tr>
<tr>
<td>QP</td>
<td>quantisation parameter</td>
</tr>
<tr>
<td>RC</td>
<td>rate-control</td>
</tr>
<tr>
<td>RED</td>
<td>random early detection</td>
</tr>
<tr>
<td>RIO</td>
<td>random early detection with in and out</td>
</tr>
<tr>
<td>RNC</td>
<td>radio network controller</td>
</tr>
<tr>
<td>RTCP</td>
<td>real-time control protocol</td>
</tr>
<tr>
<td>RVLC</td>
<td>reversible variable length codeword</td>
</tr>
<tr>
<td>SC-AIR</td>
<td>scene and channel adaptive intra refresh</td>
</tr>
<tr>
<td>SGSN</td>
<td>serving GPRS support node</td>
</tr>
<tr>
<td>SLA</td>
<td>service level agreement</td>
</tr>
<tr>
<td>SLS</td>
<td>service level specification</td>
</tr>
<tr>
<td>SQCIF</td>
<td>sub-quarter common intermediate format</td>
</tr>
<tr>
<td>TB</td>
<td>token bucket</td>
</tr>
<tr>
<td>TCA</td>
<td>traffic control agreement</td>
</tr>
<tr>
<td>TCP</td>
<td>transmission control protocol</td>
</tr>
</tbody>
</table>
TCS  traffic control specification
TM   Test Model
ToS  type of service
trTCM two rate three colour marker

UE   user equipment
UDP  user datagram protocol
UMTS Universal Mobile Telecommunication Service
UTRAN Universal Terrestrial Radio Access Network

VC   video conferencing
VFT  virtual finish time
VLC  variable length coding/codeword
VO   video object
VoD  video-on-demand
VoIP voice over IP
VOL  video object layer
VOP  video object plane
VS   video streaming

W-CDMA wideband code division multiple access
WFQ  weighted fair queueing
WRED weighted random early detection
WRR  weighted round robin
Chapter 1

1 Introduction

1.1 Background and Objectives

In recent years, the Internet has rapidly expanded and its use became immensely widespread. Furthermore, the advances in the mobile communication technologies enabled audiovisual multimedia content to be accessible by the network clients. Consequently, the requirement for effective Quality of Service (QoS) support has gained paramount importance. This has driven the need for remodelling the traditional packet network architectures to offer more efficient QoS mechanisms. On the other hand, the convergence between computing and telecommunication technologies has motivated the multimedia intensive communications to become a major issue in networking research. In wireless communications, there is a trend towards low-power, simple and multimedia capable user appliances where some of the processing and complexity loads are shifted away from the end-user terminals. In this respect, there is a need to design dynamically programmable network architectures that enable personalised and media-dependent service provision for the users.

The default service offered by IP-networks is the best-effort forwarding scheme, which was recently augmented by Integrated and Differentiated Services QoS architectures in order to support the ever growing demand for time-sensitive multimedia communications. However, neither Integrated Services nor Differentiated Services models have become widely adopted approaches. Nevertheless, Differentiated Services approach has received a great deal of attention from the research community since it has superior scaling properties, and does not need end-to-end resource reservations. Both of these models are designed to provide QoS mainly at the network level and do not incorporate specialised processing services for the multimedia traffic. To compensate for this deficiency and enhance the service quality given to the users of multimedia applications, media-gateway and proxy-model approaches were developed by research communities and commercial service providers. Nevertheless, although such approaches can ease the processing burden on the end-points and provide a level of adaptation to network impairments, suffer from scalability problems and tend to be ineffective during the increased load conditions.
Active (or Programmable) Networking is an alternative and innovative approach to the traditional packet networking architectures. It represents a significant step in the evolution of packet networks, from traditional packet-forwarding engines to service delivery entities, which enables dynamic control and management of the QoS provided for the applications. In an active networking architecture, both terminals and the network nodes are intelligent, enabling increased flexibility in service provision. In this respect, active networking technology can introduce specialised multimedia processing inside the network as well as providing support for the existing IP QoS architectures.

The specialised services deployed in an active networking environment are called the active services. In this thesis, the aim is to enhance the QoS performance of the real-time/low-latency video applications by using a series of active services. The proposed active services are visual content adaptation tools that utilise transcoding operations to perform rate-control and error-resilience computations on the compressed video streams. To achieve the pursued aim, the following steps have been followed:

- Investigation and implementation of the Differentiated Services architecture for transport level QoS provision.
- Building a low-latency transcoder for tailoring the MPEG-4 video content to the user and network requirements.
- Defining the active services.
- Testing the performance of the proposed active services over the UMTS network.
- Integration of the selected active services into the Differentiated Services architecture, and testing the QoS performance gain.

The rest of this chapter is organised as follows: The second section discusses how the performance evaluations were carried out in the thesis. The third section presents the original achievements and contributions of this research whilst the fourth section focuses on the outline of the thesis.

1.2 Quality Assessment by Performance Evaluation

The performance evaluation of the proposed active services on the video quality is accomplished with the use of two methods throughout the thesis: objective and subjective quality assessments. These two techniques were adopted due to their widespread use amongst the video coding and communication research community. In this way, the results of the experiments become comparable to the performance assessments performed by the other researchers.
Currently, the most widely used objective video distortion/quality metrics are Mean Squared Error (MSE) and Peak Signal-To-Noise Ratio (PSNR). Between the two, PSNR is the commonly adopted measurement technique that is widely being used by the research community. The mathematical representation of the PSNR is given in the following equation:

$$PSNR = 10 \log_{10} \frac{L^2}{M \times N \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [x(i,j) - \hat{x}(i,j)]^2}$$  \hspace{1cm} (1.1)$$

where $M$ and $N$ stand for the dimensions of the image, while $x$ and $\hat{x}$ are the original and distorted pixel intensity levels, and $L$ is the dynamic range of the pixel values. For an 8bits/pixel monotonic image signal, $L$ is equal to 255. Since, Quarter Common Intermediate Format (QCIF) sized video sequences were used for testing purposes throughout this research work, $M$ and $N$ correspond to 176 and 144 respectively.

Despite being a useful metric in terms of indication the levels of quality degradations in video streams, PSNR is widely accepted for not correlating well with perceived quality measurement. Therefore, it is beneficial to use subjective tests together with the PSNR measurements to get a better understanding of the degradation on the perceived video quality. The subjective quality assessment involves image comparison tests. In these tests the distorted images are compared against the original images. For most of the experiments performed in this thesis, the performance results of these two experiments are published together.

### 1.3 Source Material

Throughout the research work presented in this thesis, experiments were carried out using two standard ITU test sequences. These were the “Foreman” and “Students” sequences. These sequences are originally QCIF (Quarter Common Interchange Format) size (i.e., 176x144 pixels), in YUV 4:2:0 format, and captured at 25fps. These two sequences were used since they represent two distinctively different motion activity scenarios. While the “Foreman” sequence is highly motion-active, the “Students” sequence is an activity limited sequence. Such contrast is essential in examining the relative performance of the transcoding operations for different video applications.
Other than these standard sequences, two other sequences were used to test the performances of the transcoding algorithms. The non-standard sequences, namely the “Interview” and the “Denmark”, were captured from TV broadcasts and compressed with MPEG-1 coder at 1.5 Mbits/s. These were then resized and converted into the YUV format (i.e., 25fps with resolution of 176x144 pixels, and 4:2:0 colour format raw video) before the MPEG-4 [MPEG4] encoding process. Therefore, the raw video used in the encoder process had a certain level of noise due to the MPEG-1 [MPEG1] compression and the resolution resizing process to QCIF size.

The Interview sequence involves a slow panning and zooming camera, two logos on the top left and right corners, a banner coming into the sight towards the end of the sequence, two main subjects conducting an interview, and a passer-by in the background. On the other hand, the Denmark sequence is a football game scene, showing the highlight of a scored goal. It contains a fixed banner (the score) and the broadcast channel logo on the top right and left corners respectively. There are three scene (camera) changes in this sequence. The sequence starts with a scene in the mid-field while the camera is panning from right to the left. This is followed by a scene change where a few seconds of a close-up scene (involving two players and the ball) is shown. The final scene is the one which the goal scoring action is shown. In Figure 1.2, four frames from each of these non-standard sequences are presented. The displayed frames are chosen in such way that the aforementioned scene changes can be observed. The “Denmark” sequence is a highly active sequence with large variations in its motion activity levels. However, the “Interview” sequence is moderately active and in terms of motion activity, it can be considered to stand between the “Students” and the “Foreman” sequences.
Chapter 1. Introduction

1.4 Original Achievements and Contributions

The original research achievements can be summarised as follows:

- Development of a real-time MPEG-4 video transcoder with rate-control and error-resilience capabilities.
  - Rate-Control: Modified version of the TM5 algorithm [Tm5].
- Development of the scene and channel adaptive intra-refresh algorithm for optimum intra-block update.
- Implementation and assessment of a Differentiated Services network model for the multimedia applications.
- Development of active congestion control mechanism as a means for traffic conditioning of the video applications.

Some of the work presented here has been published in a number of international conference proceedings. A list of these publications is included in page XII. In addition, a journal and a conference paper is currently being prepared for submission.
1.5 Thesis Outline

Chapter 1 comprises an introduction to the thesis presenting the background and the rationale of the undertaken research work. In this chapter, the aim of the undertaken research work and the steps followed in achieving this aim has been defined. Further, the quality assessment techniques and the source material used throughout the thesis have been explained. After the presentation of the original achievements and contributions, the chapter ends with this section which outlines the thesis structure with brief introductory paragraphs dedicated to each of the chapters.

The main aim of Chapter 2 is to highlight the QoS service problems associated with today's networks, and draw the picture of how future networks should be designed in order to provide the ultimate QoS support for the multimedia applications. Therefore on the way to explaining the evolution of networks towards this form, typical characteristics of multimedia applications, QoS requirements of applications, transmission issues and associated problems, and the existing QoS adaptation approaches are introduced. Commonly used end-point and network based adaptation mechanisms are presented. Such mechanisms have been developed to compensate for the deficiencies of the networks in terms of delivering the required QoS for the multimedia applications. The discussions presented in this chapter are mainly focused on video-based applications like the rest of the discussions in later chapters.

Chapter 3 introduces the notion of active networking and defines its key elements. A brief discussion of how active networks emerged, why are they required, and its association with the well known concept of end-to-end arguments is presented. This is followed by the definition of architectural components, design objectives, and the programming models that have been proposed by the research community. Active networking involves introduction of processing capabilities in the network. Therefore a careful consideration should be given to the trade-off between the cost of active processing and the obtainable benefits. Another issue of concern in active processing is the granularity of control, which has also been discussed in this chapter. In the deployment of active networking components inside the networks, two important decisions have to be made; where to deploy and how to deploy. For this reason, the existing deployment strategies proposed by the research community are introduced. As far as the scope of the research work presented here is concerned, the applications of the active networks are the most important areas of investigation. Thus, a section is dedicated to present an overview of what other researches have proposed to develop using an active networking architecture. Finally the technical and presentational challenges lying ahead of active networks are discussed. The chapter end with concluding remark and the recommendations that can be made based on the investigations made on the active networking concept.
Chapter 4 discusses the Differentiated Services QoS architecture and its use for providing QoS support for multimedia applications. This architecture is capable of providing network level QoS guarantees/assurances to user applications. However, it lacks in high level application handling capabilities which can be complemented through the use of active networking technologies. In this respect, the aim of this chapter is to assess the benefits and deficiencies of Differentiated Services for multimedia applications. Therefore, the first two sections of the chapter give a brief overview of the motivation behind the development of Differentiated Services architecture, together with its service model and the approach for introducing end-to-end service provision for different traffic aggregates. The third and fourth sections provide the definition of the architectural components of typical Differentiated Networks. The traffic classification and control mechanisms are the main elements that are used to implement differentiated QoS provisioning. In Differentiated Services architecture, the service provider needs to categorise the various types of traffic that demand service from the network. Thus, an example about how different applications can be classified into distinct service classes is presented. This classification is based on the timeliness requirements of the applications. The Differentiated Services architecture has also been considered in the design of the wireless networks. Therefore particular section is dedicated for the discussion of the relation between the UMTS QoS architecture and the Differentiated Services. The rest of the chapter presents the theoretical and practical investigations performed for analysing the potential benefits of this architecture for multimedia communications.

Chapter 5 presents active services and discusses the vision in which they provide QoS support for multimedia applications. It starts with the definition of active services, and categorises them according to their functionalities. The main contribution of this chapter is the presentation of the transcoding class services. Thus, the architecture of the proposed transcoding architecture and its services are explained. Furthermore, each of the services is tested under various simulation scenarios and their performance in terms of providing QoS support for video applications is demonstrated. The complexity assessment of the transcoder is also given to prove that it is feasible to deploy such services in the networks. Finally, the active congestion control scheme is presented as an example to the possible integration of active services with the Differentiated Services network.

Finally, Chapter 6 gives the overall conclusion of the thesis, provides concluding overview of each chapter, and presents some ideas about the future research directions.
Chapter 2

2 Multimedia Communications

2.1 Overview

Multimedia communications deal with the transfer, protocols, services, and mechanisms multimedia applications transported over digital networks. The challenge in multimedia communications is how to provide adequate QoS for the transmitted data, so that the user and application requirements are satisfied without making inefficient use of network resources. In that respect, various characteristics of the multimedia applications, user expectations and requirements, as well as the capabilities of the network transport mechanisms should be carefully considered in the design of an efficient multimedia communication systems.

A myriad of research activities have been carried out in the field of enabling QoS for multimedia communications. Amongst these resource reservation and/or policy based traffic prioritisation and treatment approaches are the prominent ones. Moreover, network and/or end-user based QoS adaptation techniques have also been investigated to provide additional support for networked multimedia. In this chapter, these concepts are briefly introduced and their functionalities are discussed.

The main aim of this chapter is to draw the picture of how future networks should be designed in order to provide the ultimate QoS support for the multimedia applications. Therefore on the way to explaining the evolution of networks towards this form, typical characteristics of multimedia applications, transmission issues and associated problems, and the existing QoS adaptation approaches are introduced. The discussions presented in this chapter are mainly focused on video-based applications. In summary, this chapter establishes the background for the research work presented in the thesis.

2.2 Quality of Service

QoS is a key requirement of a distributed multimedia system, and its provision is the primary objective of the networks. Therefore, the requirements of the applications and the end-users
should be satisfied at least at an acceptable level so that the network could be considered to be economically viable. In fact, the term Quality of Service has a broad meaning and is interpreted differently by the networking and application-development communities as well as the users. From the networking point-of-view, QoS refers to being able to provide different treatment to different classes of traffic. The primary objective of providing such treatment is to improve the utility of the network. In this case, the QoS is achieved by granting priority to higher value or performance-sensitive applications over the others. The prioritisation is achieved either by dedicating some portion of network resources to specific traffic flows or by introducing preferential queueing and forwarding mechanisms in the network. Nevertheless, network QoS mechanisms cannot prevent congestion from happening but can limit the possible degradations in the application quality.

In contrast, from the application point-of-view, QoS is a mechanism that can improve the application performance. Therefore, the perception of QoS is related to the perceived quality of the multimedia application at the user-end. However, from the user point-of-view, application performance may not always be the determining factor for satisfaction. Users may require the applications to be adaptable to their perceptual requirements or dynamically tailor the multimedia content according to their terminal characteristics or environment. This situation cannot easily be associated with any of the network QoS parameters or application QoS metrics. This is the case when the network needs to employ specialised services other than the prioritised transportation mechanisms where certain signal processing operations are performed on the multimedia applications prior to their arrival at the user terminal. Popular examples to such adaptation mechanisms are introduced in Section 2.5.

Although the term QoS has multiple meanings, a general definition is made by Vogel et al [VogeA] and is the understanding adopted in this thesis. This definition is:

"Quality of Service represents the set of those quantitative and qualitative characteristics of a distributed multimedia system necessary to achieve the required functionality of an application"

Based on this definition Quality of Service can be thought to represent a set which contains parameters representing different levels of QoS. Depending on the multimedia type, this set can contain different parameters. In [WolfL], four distinct levels of QoS were identified. These are:

- **User QoS**: The user QoS parameters describe requirements for the perception of the multimedia data at the user interface.
• **Application QoS**: The application QoS parameters describe requirements for the application services, possibly specified in terms of media quality (such as end-to-end delay) and media relations (such as inter-media synchronisation).

• **System QoS**: The system QoS describe the requirements from the communication services resulting from the application QoS. These may be specified in terms of both quantitative (such as bits-per-second or task processing time) and qualitative (such as multicast, inter-stream synchronization, error recovery).

• **Network QoS**: The network QoS parameters describe the requirements on the network services (such as network load or network performance).

The end-to-end QoS in multimedia communications can only be attained if all of these four levels of parameters are compiled together to produce an overall service behaviour. Based on the arguments given in [VogeA], an end-to-end QoS behaviour can be produced using the following steps.

1. Assessing the QoS requirements in terms of users’ subjective wishes or satisfaction with the quality of the application.

2. Mapping the assessment results onto QoS parameters for various system components or layers. For example, the user chooses the video in terms of its resolution and frame-rate, which maps onto throughput requirements.

3. Negotiating between system components or layers to ensure that all system components can meet the required parameters consistently.

The system can agree to grant the necessary measures to satisfy the end-user or may reject the QoS request. The negotiation takes place prior to the start of media transmission through the use of some sort of signalling protocol (e.g., as in H.323 [H323] and SIP [RoseJ]). Usually, the QoS parameters can be renegotiated at any stage of the communication if the network conditions alter or the user decides to pay more for better quality.

To generalize, these four distinct levels can be grouped in two classes; the QoS parameters that can be subject to negotiation (User QoS and Application QoS) and the nonnegotiable parameters (System QoS, Network QoS) between system components. That is to say, whether a bit-stream should be multicast or not has no relevance with the end-user’s perception of media quality. In this context the negotiation refers to the statement of necessary actions to be taken to acquire the required perceptual satisfaction.
Chapter 2. Multimedia Communications

2.2.1 Network QoS Parameters

Packet networks are represented by layered reference models. The number of layers in every reference model and layer names can vary depending on the network model being used [TaneA]. However, there are four inherent layers that are vital for QoS provisioning. Each layer of the network has different tasks, which in turn contributes to a different aspect of QoS notion. In fact, the services provided by each layer complement the one above. This implies that the QoS provisioning cannot be regarded as a single layer service but as a combined service of all layers involved in the communications process. The main layers and their functions are as follows:

- **Physical Layer:** Physical layer operations, such as the use of specific channel coding scheme, forward error correction, and automatic repeat request techniques have significant effect on the multimedia application’s performance.

- **Data-Link Layer:** This layer responsible for error-detection/correction and flow control. In wireless networks, this layer also performs the power control. In this respect, the data link layer is very important since it has a direct effect on the user’s perception of media quality.

- **Network Layer:** The network layer protocols are responsible for routing, congestion control, seamless interconnection and mobility management. For example, in the case of network congestion, the routing tables can be altered to reduce the traffic load on a congested link, hence reducing the probability of packet loss.

- **Transport Layer:** Transport layer services allow users to specify the user service requirements such as preferred, acceptable or minimum values for service parameters. These values are then negotiated with the network servers. For instance, resource reservation signalling is initiated at the transport level where the demand for bandwidth and other resources required by the service are negotiated.

- **Application Layer:** At the application level various support services and content adaptation tools can be provided to support application QoS. For example, using the functionalities within the application layer, real-time services can be made adaptive to changing network conditions, (e.g., enabling error-resiliency options in the encoder). Security issues such as encryption and authentication are also related to the application level.

Overall, the effects of different communication layers can be modelled with the use of network QoS parameters. From a network-centric point-of-view, the most important parameters that are used to characterise the performance of an IP network and influence the end-to-end quality of an application are:
Chapter 2. Multimedia Communications

- **Bandwidth**: It represents the portion of the network capacity that is available to the application traffic. In other words, bandwidth determines the maximum allowable traffic volume that can be transmitted between network end-points. Depending on the network QoS architecture in use, particular applications can have a reserved portion of total network bandwidth or need to share it with other applications.

- **Packet Loss**: In fixed networks, packet loss is typically the result of congestion in the network. In wireless networks, packet loss is mainly associated with the corruption of the received packets due to the channel noise. Generally, packet loss is defined as the percentage of IP data packets, out of the total number of transmitted packets that are lost somewhere in the path from the source to the destination [AlmeG]. For multimedia applications, the packet loss percentage itself is not always enough to assess the impact it has on the perceptual quality. In such cases the pattern of loss is also a significant parameter. Nevertheless, service providers usually consider it as one of the important parameters which defines the QoS offered for a particular aggregate of application traffic.

- **The Loss Pattern**: The packet losses in today’s IP networks are typically bursty but the burst duration can show variations. Depending on the congestion management strategies employed, the burst length can be minimised [RamaK]. The loss pattern may cause minor distortions or significant deterioration of depending on the encoding mechanism and the error correction as well as concealment techniques being used. There has been a considerable amount of measurement and research aimed at characterizing and modelling packet loss patterns in the IP networks [BoloJ], [BoreM], [YajnM].

- **Delay**: Delay is the time taken for application data units (packets) to be transported by the network to the destination. Delay is caused by a number of factors including propagation delay, processing delays, and queueing delays at the intermediate routers on the path to the destination host [AlmeG]. If the delay is above certain extent (depending on the application) the intelligibility of real-time interaction can be degraded and sometimes become unusable for the users.

- **Delay Variation**: Delay variation is also known as *jitter* and is usually caused by the buffer build-up on routers during periods of increased traffic, weighted scheduling mechanisms, and due to routing. Multimedia applications with stringent delay
requirements need the delay variation to be as little as possible. Delay variations are usually smoothed out using buffers at the end-points. However, the tolerance of the application to the buffering is limited since buffering contributes to the delay.

2.2.2 The User Perspective

Multimedia QoS is typically measured using technical parameters such as delay, delay variation, packet loss, and throughput. These parameters are known to be useful for both user and network aspects of QoS provisioning. However, these parameters are not necessarily adequate in terms of representing the requirements of users given the restrictions imposed by the network resources, admission policy, and different psychological factors which affect the QoS perception of users. Users, require QoS models that are tailored to their needs, and that are expressed by different performance characteristics such as response time, predictability, and consistent perceptual quality [MiraD]. The human element in multimedia communications, although appreciated, has often been overlooked. It is argued that, unless the human perceptual mechanisms are taken into consideration while designing any QoS architecture, it would be wrong to assume that the QoS techniques could match to user requirements and provide satisfaction.

There is a direct relation between end-user expectations and requirements, and advancements in the technology. People adapt themselves to the innovations with an incredible speed and once services of better quality are available, they do not feel satisfied with the older services which they have been using before. Therefore, understanding of the user requirements is a necessity in designing a system that will operate efficiently (i.e., in terms of using network resources), and effectively (i.e., providing satisfaction for end-users) in the distributed multimedia environment. Investigations have revealed that there exists a threshold beyond which the users do not perceive any improvement in the media quality no matter how much network resources are dedicated to the application [ApteR], [FukuK].

The human perceptual mechanism is a non-linear and a complex system. Consequently, there is an inherent difficulty and subjectivity associated with the understanding of one's sense of multimedia perception [GhinG]. In other words, perception is related with psychological factors and the quality evaluation of the same media may show differences between multiple experiments, if the user’s psychological condition varies in time. The temperamental attitude of users can be controlled by using a different pricing scheme for different QoS levels and letting them know what kind of quality they should expect in near future while they are involved in an ongoing multimedia session [BoucA].
Chapter 2. Multimedia Communications

The research work presented in [BoucA] suggests that users' assessment of the value of QoS received is influenced by a number of different factors, and hence the same level of quality may be perceived differently under different circumstances. It is claimed that, users require, rather than networks ability to provide higher QoS levels, a predictable level of QoS that allows them to make accurate value judgments about the quality they receive. As well as being predictable, QoS should also be consistent. That is to say, a reasonably lower and consistent QoS can be rated higher than the higher but inconsistent (showing large quality fluctuations in an unpredictable manner over the time) QoS. The quality prediction can be enabled by network sending feedback signals to the user, informing about the near future QoS that should be expected. However, the feedback should only be provided if there is going to be a considerable change in the pre-negotiated quality level, hence should not distract the user from the primary task. It is suggested that the network should only signal when there is a relatively high cost in using the network resources, and there is a high cost placed on keeping the current QoS level using the pre-negotiated task value. In conjunction with these, it is essential to implement a pricing scheme for different levels of QoS that can be provided to the user by the network service provider. In this way, the users' expectancies of QoS can be limited and satisfied at lower cost (in terms of network resources). It is therefore essential to investigate the factors affecting human perception of multimedia data, especially audio and video. For this purpose, a perceptual model should be studied and the dependency of different media types on each other should be defined. The ultimate aim in doing this is to find a way of mapping the user requirements onto network QoS parameters.

2.2.3 Application QoS Metrics

The effects of QoS enabled treatment on multimedia applications are usually measured in terms of user-perceivable effects. Although network-centric QoS parameters play an important role on the application performance, the effects they have on different applications or the variations of the same application are inherently dissimilar. For example, the perceptual quality degradation caused by a certain level of packet-loss on two different Voice-over-IP (VoIP) applications that are encoded at the same bit-rate would not be the same due to the differences in compression algorithms used. Thus, it is not possible to directly map these parameters onto the application quality.

The application quality is therefore defined with a set of metrics which are affected from the effects of network-centric QoS parameters as well as the performance of the algorithms used to create the multimedia content. The application QoS metrics are:
Chapter 2. Multimedia Communications

- **Latency:** This is the end-to-end delay that the application experiences. Latency has a direct influence on the user's perception of quality. If latency exceeds the limits defined by the application, it can affect the natural play-out of the multimedia content and become perceptually disturbing.

- **Bandwidth:** Applications demand a specific amount of bandwidth from the network throughout their session in order to deliver the content at the intended quality. If minimum required bandwidth is not available, the real-time delivery becomes impossible.

- **Data Loss:** Data loss refers to packet losses experienced in the congested or error-prone channels. This results in erroneous decoding of the received information and in some cases drastic degradations in the perceived quality. However, depending on the robustness of codec being used and the amount of information lost, data losses may not always be perceivable to the users. In addition, the effect of data loss on the perceptual quality is also related to the content of the application. For example, 1% packet loss may have little effect on a low motion active MPEG-4 video stream, while the same loss ratio may be more significant for a high motion active video stream. Moreover, the occurrence pattern of the data loss is another determining factor in perceived multimedia quality. Data loss can also be associated with the encoding parameters used at the multimedia source. If excessive compression is used, the intelligibility of the multimedia content can be regarded unsatisfactory by the user.

- **Jitter:** The effect of jitter can usually be alleviated by the end-application through the use of buffering. However, if jitter becomes exceedingly high, some packets may arrive at the destination too late to be useful. In this case, late packets are regarded as lost packets. The application may wait for the late packets which cause temporal inconsistency in the presentation of the application (e.g., video image freezes and then starts again). For real-time multimedia applications which involve two-way interactions, the perceptual disturbance caused by the jitter is worst than having a constant delay. However, if the applications are not extremely time-critical (e.g., as in streaming multimedia) jitter can be removed by means of application buffering but this would increase the end-to-end delay.

### 2.3 Classification and Requirements of Multimedia Applications

It is commonly acknowledged that most of the advanced multimedia applications cannot be entirely accommodated by today’s best-effort IP networks, thus it is necessary to have a service model which can provide QoS based traffic handling. The opposite school of thought claims that
QoS needs of applications can be satisfied by over-provisioning the existing networks (i.e., make more bandwidth available), and developing intelligent applications that can adapt to the changing network resources. However, in the view of the research work presented in this thesis, the effective solution lies somewhere between these two approaches. While it is obvious that adaptive multimedia applications would make better use of network resources, applications would certainly benefit from some form of traffic differentiation. Without such measures, multimedia applications are likely to suffer from network congestion which can have adverse effects on QoS-sensitive network traffic. As a result, traffic flows will be subjected to packet drops and increased end-to-end delays which effectively reduce the presentational quality of the corresponding multimedia applications.

In order to design an effective QoS enabled transport network, the requirements of the applications, users, and the availability of network resources should be considered in conjunction to each other. Since networks are used by end-users running certain applications on their terminals, it is important that the network operators and service providers consider the requirements of those applications transmitted over the network. In addition, the effects of available network resources and the performance of the employed QoS mechanism on the application performance should also be calculated. On the other hand, applications are required to make efficient use of network resources and be robust against possible bandwidth fluctuations. Applications that cannot respond to the changing network conditions can overtake the network resources and result in network congestion or even congestion collapse [FloyS2], and reduce network utilisation.

Therefore, understanding the characteristics and the requirements of multimedia applications is essential, since this can be used to tailor network services that suit to the demands of such applications. On the other hand, application and service developers can make use of these characteristics to improve application response to the adversities associated with the networks. In the similar way, understanding application needs and network conditions can allow applications to deploy/initiate built-in/network-based mechanisms that could allow them to function with the desired quality even when the network conditions or user requirements change over the time. Nevertheless, networks are not usually designed to provide specialised services for every possible application that might be used by the end-users. Therefore, applications can be classified into different groups which contain applications of similar properties and requirements. In this respect, applications can be categorised based on their presentational and networking characteristics. While the presentational characteristics define the properties of multimedia applications from the user's point-of-view, the networking characteristics define the differences of traffic behaviour amongst different applications.
2.3.1 Presentational Characteristics

Multimedia applications can be categorized as real-time and non-real-time, meaning to say whether the media has a time dimension or not. Within the real-time applications further classification can be made depending on whether the applications are presentational or conversational, although most applications can have both presentational and conversational aspects [VogeA]. For example, streaming media services are categorized as presentational while teleconferencing falls into conversational category. In the case of presentational applications, the media stream flows in one direction in the network and at the end-point some buffering (i.e. a few seconds) is used to provide a smooth and relatively continuous presentation. On the other hand, conversational applications, unlike the case with streaming applications, has got more stringent requirements in terms of timely delivery, and the delay introduced should not cause any perceptual disturbance to the end-user. Therefore, the distinction between presentational and conversational applications is required in order to be able to precisely determine the characteristics and the basic requirements of the multimedia applications.

Most audiovisual multimedia applications are time-sensitive, involve a level of interaction, and require low-latency or real-time communication. Hence, these types of applications have very stringent delay and delay jitter requirements to preserve the interactive nature of their content. In a multimedia system, the sum of all delays in each component and process of the network is called the end-to-end delay. In general this includes, disk access, analogue to digital conversion, encoding, host processing, network access, network transmission, buffering, decoding and digital to analogue conversion. The concept of acceptable delay is a very subjective matter since the maximum end-to-end delay requirement is usually application dependent. It is widely accepted that the end-to-end delay for a conversational type of applications should not exceed 300ms [HenmD]. However, streaming media is not that demanding and for most applications, a response time of a few seconds is acceptable [LuG].

Due to the nature of the packet switched networks the delay introduced in the network is not constant. In a distributed multimedia operation environment, data packets do not arrive at the destination in fixed intervals as required by the audio/video applications. The main reason for this is the delay variations introduced during the network's data transmission and processing times. As explained previously, the variation in the delay is called the jitter. The most straightforward way of dealing with the effects of the delay jitter is to buffer the incoming data packets for an acceptable period of time, determined by the characteristics of the multimedia session, prior to decoding and playing.
2.3.2 Networking Characteristics

User applications can also be classified based on their networking characteristics. Networking characteristics refer to application properties attributed to the nature and transport requirements of the corresponding traffic flows. In this respect, applications’ traffic can be classified as elastic/inelastic and tolerant/intolerant [MiraD].

Applications with elastic traffic can tolerate significantly high delay, jitter, and variations in bandwidth without being significantly affected. These are traditional data transfer applications, which use reliable transport mechanisms (i.e., Transport Control Protocol (TCP)). While fluctuations in the network QoS parameters may degrade their performance (e.g., in terms of delay experienced), elastic applications are not affected by unfavourable network conditions. However, some of these applications can be more sensitive to such adversities. Thus, applications with elastic traffic can further be classified according to their delay and throughput requirements. For instance, latency and throughput requirements of applications such as email and voice-mail are very relaxed. On the other hand, applications such as interactive web, haptic-device communications, and net-auctions, which involve human-to-machine interaction, should have delays of 200-300ms or less, but could possibly accept slightly more.

Applications with inelastic traffic are called real-time/low-latency applications which usually involve audiovisual multimedia content. Such applications are sensitive to delay, jitter, bandwidth fluctuations, and errors. Therefore, if certain QoS provisions are not in place, the perceived quality such applications may rapidly deteriorate, and may even become unacceptable for the users. As in the case with applications with elastic traffic, real-time/low-latency applications also show diversities in terms of their tolerance to network adversities. For example, streaming applications (i.e., presentational multimedia), unlike conversational multimedia, are loosely interactive, and not extremely sensitive to delay and jitter. Thus, applications can further be classified according to their tolerance levels to network diversities (i.e., tolerant versus intolerant).

Multimedia applications are usually somewhat tolerant and can operate within a range of QoS provisions with acceptable or satisfactory quality. The content of such applications are created in such a way that up to certain level of channel impairments can be concealed or smoothened for the human perceptual system. Tolerance can also be provided by enabling the applications to be adaptive to changing channel conditions. In general terms, adaptation can be introduced, for instance, by utilising de-jittering buffers and regulating the application bit-rate to account for the network congestion. However, some applications do not possess such built-in adaptation mechanisms, yet can still provide limited resilience to possible fluctuations in the network.
resources. For instance, the quality of an audio or video flow may be degraded by packet loss, but they can still be intelligible to the user. On the other hand, intolerant applications are unable to accomplish their task successfully if certain QoS provisions are not guaranteed. An example of such application is the remote control of mission critical equipment, such as control of a robot or surgery instruments. Intolerant applications can as well be adaptive or non-adaptive.

It should be noted here that the assessment of QoS by the user is shaped by various factors other than intelligibility, and involves psychological elements as well as the usage context of the application. Therefore, the adaptation to network impairments as well as the user requirements ideally should include such factors as an inherent component of the QoS provisioning system. The utmost challenge in multimedia communications is therefore the provision of adaptation mechanisms which can make balanced consideration to all elements of QoS perception.

2.3.3 Voice-Based Applications

Audio applications refer to a wide range of applications including VoIP, voice/audio streaming (e.g., as in Internet radio), and audio orchestration. VoIP and streaming applications are perhaps the most commonly used multimedia applications in today's Internet. Moreover, IP based voice applications are also likely to become an integral part of wireless multimedia communications over 3G networks and beyond. In this thesis, the discussion is concentrated on to voice-based applications and VoIP in particular. Popular VoIP codecs used in the Internet are G.723.1 [G723], G.729.A [G279], and iLBC [AndeS].

2.3.3.1 QoS Requirements for VoIP

VoIP has stringent requirements regarding end-to-end delay, jitter and packet loss. Typical bit-rates for VoIP applications range from a few kbits/s to 64kbits/s. Although their transmission rates are not substantial, VoIP applications are presentational type multimedia and require sustained bandwidth for intelligible communications. However, bandwidth provision may not be sufficient to satisfy the perceived quality by the users. From the user point of view, the perceived voice quality is assessed based on a number of factors including fidelity, intelligibility, delay, and echo [MiraD]. In this section, the quantitative analysis of the effects of delay, jitter, and packet loss on the perceived quality of the VoIP applications are introduced.

Delay mainly affects conversational quality rather than perceived voice fidelity. Based on the research work of International Telecommunication Union (ITU), the users cannot notice any delay below 100-150ms [G114]. If the delay is between 150 and 300ms, it is causes as a slight hesitation in the response of the parties in conversation. However, the talker-overlap (the problem
of one caller stepping over the other talker's speech) effect becomes perceptually disturbing when
one-way delay is around 250ms. If the delay above 300ms, it becomes obvious to the users, and
real-time two-way conversation becomes impossible. As a result, the maximum one-way delay
experienced by VoIP applications ideally should not exceed 150ms.

Excessive jitter introduces jumble in the conversation. In general, jitter should be less than 50ms
[KaraJ]. De-jitter buffer can be used to alleviate the jitter effect. Jitter buffers sizes can be variable
and adaptive to instantaneous network jitter conditions. However, there is an upper limit to
acceptable jitter beyond which late packets are either discarded or considered lost. The packet loss
may introduce conversational gaps and result in a confusing conversation in which the talking
parties may jumble together.

The effect of pack loss on the perceived quality of VoIP depends on a number of factors: the
speech coder used, the existence of error protection or correction schemes (e.g., Like Forward
Error Correction (FEC)), and the pattern of packet loss. In general, voice codecs cope better with
packet loss events where the packets are lost individually rather then in bursts. This is because
error-resilience mechanisms built in the codecs are usually effective in recovering from isolated
lost packets, but may be unable to cope with a lengthy series of consecutive packet loss.
Furthermore, the effect of packet loss also depends on the packet size. When smaller packet sizes
are used (e.g., 20-30ms), the adverse effects of packet loss can be alleviated by using the error
concealment mechanisms employed in the coders. However, when the packet sizes are larger
(80ms and above), it becomes more difficult to interpolate the lost information from the received
packets. Nevertheless, small packet sizes increase the bandwidth overhead due to the transport
and network layer overheads. Therefore, selection of the optimum packet size constitutes a trade­
off between the specific speech coder, the impact of packet loss, and the packet header overhead.
The location of loss within the bit-stream is also another important factor affecting the perceived
quality. For instance, the loss at an unvoiced speech segment is less significant in terms of the
perceived quality then the loss of a voiced segment [SunL].

When speech signals are transmitted over noisy channels (e.g., as in wireless channels), bit-errors
can be introduced which results in corruption of the packets and consequently misinterpretation of
the decoded information. Moreover, when the transport or network layer headers of a packet are
corrupted, the packet is dropped at the decoder and considered to be lost.

In [MiraD], it is argued that the experiences from the tremendous acceptance of mobile or cellular
telephony (i.e., GSM) has shown that users are prepared to tolerate transient degradations in the
voice quality, provided that the appropriate incentives are in place (e.g., it is the preferred means
for performing voice communication). However, it should be noted that the context of use (criticality of application, quality expectations, and other psychological factors) is still the most influential factor in the perception of the voice quality.

2.3.4 Video-Based Applications

Video-based applications can be generalised into either video-conferencing or streaming types. Some examples to video-conferencing type applications are video telephony, virtual collaborative environments, distance learning, and remote surgery. On the other hand, live video transmission, multimedia broadcasting, Pay-TV, and Video-on-Demand (VoD) are examples to streaming type applications. A wide range of bit-rates, resolutions, frame rates, and video codecs are used in creating these applications. MPEG-1 [MPEG1], MPEG-2 [MPEG2], MPEG-4 [MPEG4], H.261 [H261], H.263 [H263], and H.264 [H264] are amongst the most popular video codecs used.

Both types of applications can be transmitted on one-to-one or one-to-many basis. In today’s Internet and wireless networks (e.g., 3G), these applications are usually restricted to low/modest bit-rates that range from 32-500kbits/s. This is due to the limited bandwidth in backbone and access networks, lack of QoS support, and unpredictable behaviour of transport channels. However, high-bandwidth applications (e.g., in the range of Mbits/s) are also being used in dedicated networks. In this thesis, the focus is on low bit-rate video communications.

The perceived video quality at the user end is a complex function of several factors: the bit-rate, frame rate, resolution of the video image, the context of the application usage, and the expectations of the users etc. The end-to-end delay and jitter are important parameters affecting the application's interactivity and the timely delivery of video information for decoding operation, and packet loss can have detrimental effects on the visual quality. Both video-conferencing and streaming type applications have different requirements on delay, jitter, bandwidth, and packet loss. The following subsection highlights these issues and provides a quantitative analysis of their impact on video communications. Although the effects for each of these parameters are presented individually, it should be mentioned here that the perception of video quality at the user end is a cumulative outcome of the performance of these parameters. However, studying the effects of these parameters individually is a more tractable approach in terms of understanding how they influence the perception of video quality. By this way, the applications and the networks can be improved to provide better support for QoS enabled communications.
2.3.4.1 QoS Requirements for Video-Based Applications

Video-conferencing type applications are interactive in nature and therefore have stringent delay requirements. Generally, one-way delay should be less than 600ms [MiraD]. This figure is considerably larger than the delay requirement of VoIP applications. This is because, the visual aspect of video-conferencing applications help users to synchronise their actions with the other conversation party. However, in mission critical applications (e.g., remote surgery), the delay should be lower such that the user actions correspond to events taking place on the remote location. In the case of excessive delay and jitter, the display will pause temporarily, resulting in display discontinuity and directly effecting the user's perception of the video quality.

Streaming applications, on the other hand, are limited in interactivity. Therefore, their delay requirements are far less stringent than video-conferencing applications, and can tolerate a start-up delay in the order of seconds (i.e., typically less than 10s). Streaming applications use long buffers to store a few seconds of the application clip. This provides resilience to any variation in packet inter-arrival times and transient congestion in the network. Nonetheless, the transmission delay should be limited in order to constrain jitter to maximum 500ms [UKER].

In general, the video-conferencing applications are more flexible in the throughput requirements and they are able to tolerate to higher distortion than the video-streaming applications. This is because, in video-conferencing the primary task is to communicate with the opposite party and the content of the application (e.g., the details in the speaker's background) having secondary importance. However, in video-streaming applications the interactivity is lower and the sustained quality of the video is more significant (i.e., sustained bandwidth is essential). Therefore, video-conferencing applications are more tolerant to packet-loss and can utilise rate-adaptation techniques to regulate their bandwidth requirements. It was reported in [UKER] that video conferencing applications can tolerate up to 5% packet-loss, where as this should not exceed 2-3% for video-streaming applications [MiraD]. In fact, it is difficult to state a particular percentage of packet-loss rates are acceptable or unacceptable for video applications. The effect of a specific packet loss rate on the video quality depends on the redundancy techniques used (i.e., Automatic Repeat Request (ARQ) and Forward FEC) [BoloJ2], [RheeJ], error-resilience [TallR] and concealment mechanisms employed [WangY], and the expectations of the users [BoucA].

Video-based applications usually employ Variable Length Codes (VLC) in order to achieve high compression gains [TekaA]. VLCs are very sensitive to bit errors since a single bit error can propagate into many VLCs [SadkA]. While bit-errors are rare in fixed networks, they are usually associated with wireless networks, and can be detected at the transport level or at the source
coding level. At the transport level, error detection is normally performed on packet level. If the transport headers are corrupted then the packet is considered lost. However, if the error is located inside the video data then the video decoder may decide to drop a video segment that contains errors, or use the wrongly decoded information to uncompress the video data. The level of bit corruption is measured with Bit-Error-Rate (BER). The performance of video applications under a certain BER can vary depending on the application characteristics, error-resilience [TallR] [SadkA], concealment [WangY], and channel coding mechanisms used (e.g., turbo codes and convolutional codes) etc.

The jitter requirements of video-streaming applications, like their delay requirement, are relax and up to 500ms of jitter is usually acceptable. However, video-conferencing applications cannot afford to have long buffers to absorb the delay variations in the packet inter-arrival times. Consequently, the jitter requirements of such applications are considerably lower than those of video-streaming, yet it is difficult to specify specific values since jitter depends on the application bit-rate, packet sizes, and packet inter-arrival times.

Video-based applications are usually encoded together with other multimedia content such as audio. Therefore, interaction between different media types in video-based applications is another important factor that has an impact on the perceived presentational quality. That is to say, it is essential to preserve the temporal synchronism between the video and audio streams of the same audiovisual application. When media flows lose their inter-stream synchronisation, the users perceive the application as artificial, awkward and annoying [SteiR]. To prevent this, advanced multimedia standards like MPEG-4, utilise multiplexing strategies to encapsulate the interdependent audio and video streams into the same packets. However, avoiding multiplexing of the media streams can provide with more chances of applying different transmission, adaptation and error protection strategies to the individual flows. For example, in the case of transient network congestion, audio packets can be prioritised over the video packets in order to preserve the continuity of the conversation. The cause of synchronisation loss is usually associated with packet loss and jitter, although this can also take place during the media creation process (i.e., due to wrong configuration of the capturing and encoding devices). Studies have shown that a lag of \( \pm 80\)ms between the audio and video flows in a multimedia presentation is an acceptable level for preserving the synchronisation [SteiR].

Based on the investigations carried out with a number of video codecs, ITU has compiled the information shown in Table 2.1 as an indication of suitable performance targets for video-based applications over the Internet. As can be seen in the table, in comparison to the figures reported in [UKER], the ITU findings show differences in PLR and delay requirements for video
conferencing and video streaming applications. This is due to the use of H.323 codecs in [UKER] experiments.

<table>
<thead>
<tr>
<th>Application</th>
<th>Degree of Symmetry</th>
<th>Typical Data Rates</th>
<th>Key Performance Parameters and Performance Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video Conferencing</td>
<td>Two-way</td>
<td>16-384kbits/s</td>
<td>One-way Delay: &lt; 150 ms preferred &lt;400 ms limit</td>
</tr>
<tr>
<td>Video Streaming</td>
<td>One-way</td>
<td>16-384kbits/s</td>
<td>&lt; 10 s</td>
</tr>
</tbody>
</table>

Table 2.1: Indication of Suitable Performance Targets for Video-Based Applications [G1010]

2.3.4.2 Effects of Packet Loss and Bit-Errors on the Video Quality

Most video codecs reduce spatio-temporal redundancies by using quantisation and inter-frame compression, which is based on motion compensation and estimation. If the packet loss affects a predictive frame (P-frame), then the errors introduced will propagate to all future predictive frames until an error-free intra-frame is encountered. This would cause severe distortion and rapid degradation in the perceived video quality.

The amount of error introduced due to packet loss is related with the packet sizes, spatio-temporal location of packets within the video sequence, and the amount of existing redundancy in the video sequence. A packet loss may affect only a small spatial location of a video frame or can lead to the loss of complete video frame(s), depending on the size of the packet and the importance of the lost information (e.g., if frame header is lost then the whole frame would need to be dropped). If packet loss pattern is bursty, it may result in the loss of many video frames. Therefore, the choice of the right packet size is an important choice. The impact of a lost packet is higher if the packet size is large. On the other hand, smaller packets will cause less distortion but, would mean that more packets will have to be transmitted. This, however, increases the header and packetisation overhead and decreases the throughput.

When the packet loss affects only a limited spatial location within a video frame, error concealment may be applied to the missing video data [WangY]. If complete frames are lost, the decoder either replaces it with the last correctly received frame, or tries to use temporal frame interpolation to keep an adequate frame rate.
decoder either replaces it with the last correctly received frame, or tries to use temporal frame interpolation to keep an adequate frame rate.

Wireless channels are noisy and introduce bit-errors and burst errors in the compressed video stream due to fading and multi-path reflections [SklaB]. Bit-errors can also cause packet-loss as explained in the previous section. However, due to the characteristic of VLCs a single bit error in the video stream may not be localized to a small spatial region, and can lead to the corruption or loss of many macroblocks due to the loss of synchronization between the decoder and the bit-stream. When this happens, the decoder will discard all the video information (whether corrupted or not) until the recovery of next correctly decoded resynchronisation word. This means that the information loss can range from a single macroblock to a number of frames.

Particularly interesting is the interaction between the throughput and packet loss. Under normal circumstances, the quality of an encoded video stream increases with the reduced compression ratio (i.e., increasing bit-rate). However, the research work presented in [VersO] and [WangY] shows that when packet losses (or bit-errors) are present, quality increases up to a certain bit-rate, but then starts to decrease smoothly. This is because, at lower bit-rates, the distortion on the video quality is associated with the high compression rates, and therefore the quality is improved with the increasing bit-rate. However, after a certain bit-rate (depending on the video content), the quality saturates and then starts to drop since a greater number of packets would be lost. Consequently, the distortion introduced causes larger visual quality degradation. Therefore it can be claimed that, given a certain packet loss rate, there is an optimal average bit-rate, which is dependent on the video content [MiraD]. This issue is further elaborated in Chapter 4.

In conclusion, that there is a need for designing network adaptation mechanisms which can modify the application bit-rate based on the congestion levels in the networks. Whether these mechanisms should be employed at the end-user or specialised network nodes is the subject of discussion for Section 2.5 and Section 2.6.

2.3.4.3 ISO MPEG-4

MPEG4 is the state-of-the-art international standard for coding of Audiovisual Objects (AVOs). In other words, MPEG-4 is an object-based multimedia architecture which allows for the separate coding of AVOs. It defines a vast set of compression technologies and formats that address a wide range of applications. MPEG4 standardizes storage file formats and carriage of rich media over a broad range of narrowband and broadband transport networks and supports playback on a variety of platforms and devices. Therefore, MPEG-4 is a generic name for a comprehensive multimedia
coding and communication standard. However, the discussion of MPEG-4 in this thesis is limited to its visual aspects as a video coding standard.

The aim of the MPEG-4 standard was to achieve very low bit rates by efficient compression algorithms, particularly for mobile and wireless multimedia communications. Initially, MPEG-4 was targeted primarily at very low bit rate video communications (5kbits/s to 4Mbits/s). However, its scope was later expanded to become a multimedia coding standard. Although the standard is not targeted for any specific application, it supports variety of functionalities which may be utilised by many different applications [MPEG4]. Its error-resilience properties are especially well suited for applications transmitted over mobile and wireless networks [TallIR].

MPEG-4 has been designed to provide solutions for the following issues:

- **Interoperability**: The MPEG-4 standard is not designed to target a specific platform. Instead, it aims at interoperability and can operate on all platforms.
- **Transport Independence**: MPEG-4 does not specify any requirements in terms of the transport technology to be used. Thus, content providers can tailor their applications to be used in a wide range of networking environments.
- **Improved Coding Efficiency**: MPEG-4 was targeted primarily at very low bit-rate compression (5kbits/s to 4Mbits/s) of rich media streams with improved coding efficiency in comparison to its predecessor standards.
- **Universal Access**: Robust coding tools allow MPEG-4 to be accessible over a wide range of users in mobile networks as well as wired connections.
- **Interactivity**: MPEG-4 enables the users to manipulate the scene description and the properties of the AVOs.
- **Content-Based Scalability**: AVO based coding enables content-based applications. Object manipulation, bit-stream editing, and object-based scalability allow new levels of content interactivity in MPEG-4. For instance, the bit-rate and resolution of the video object can be adapted to the different requirements of the users.
- **Profiles**: The use of profiles allows the content creators to implement only a specific part of the standard that they need. This limits the tool set a decoder has to implement and reduce the computational complexity. For the research work reported in this thesis, only the simple visual profile was used.

MPEG-4 uses a hierarchical approach in describing a scene. Each scene is defined by Video Object (VO), Video Object Layer (VOL) and Video Object Plane (VOP) lying on top of each other hierarchically. A VO is the basic element within a video scene and can be of arbitrary shape.
There can be a number of VOs in a scene or there can be only one. VOP is the encoded form of the video object, and when there is only one object in the scene VOP corresponds to a single frame. The encoding and decoding processes in MPEG-4 are carried out on the instances of the VOs which constitute the VOPs.

Spatial scalability and temporal scalability are also supported in MPEG-4. Object-based scalability can be achieved by means of layers known as VOLs which represent either the base layer or enhancement layers of a VOP. The layered coding of the video information allows adaptation of the multimedia stream to different network conditions and user requirements. In layered coding, base-layer represents the minimum level of video quality, and its quality is increased with the addition of the enhancement layers.

In this thesis, only the basic profile of the MPEG-4 codec has been used. The investigated usage scenarios mainly included end-users accessing the video content through restricted resources in terms of bandwidth and/or terminal capabilities. Therefore, MPEG-4 was chosen because of its efficient compression and advanced error-resilience features.

2.4 Multimedia Transport over Wired and Wireless Networks

The Internet Protocol is the network layer which provides interoperability amongst different networks and devices. IP has become the common interface of both wireless (IP-based) and wired (fixed) networks. In this sense, there is no difference between the wired and wireless networks if only network and higher level protocol layers are concerned. It is the protocol layers residing below the network layer that creates the difference amongst networks. Although there are networks which operate based on reliable connections, the scope of the research work presented here is limited to the IP based architectures only.

The adversities experienced by the users in terms of QoS in wired networks comes from the limitations introduced by the access network bandwidth and the congestion happening due to inadequate QoS support. However, for the wireless networks congestion is not a primary concern but the bandwidth available to individual users, noisy channel conditions, and interruption in service due to handoff or disconnection are the main hampering factors for satisfactory QoS provision. The other important issue that concerns both types of networks is the heterogeneity existing in network access technologies, user terminals, applications, network management policies and so on. Therefore, effective mechanisms should be developed to provide seamless and adaptive QoS provisioning across the networks.
Chapter 2. Multimedia Communications

The aim of this section is to introduce the multimedia transmission aspects of fixed and wireless networks. For this purpose, QoS architectures and multimedia communication protocols of these networks are presented.

2.4.1 Communication Protocols

Packet networks employ a large set of standardised protocols for enabling traffic forwarding, service provisioning, and network management functions. In this section those protocols that are related to the transport of multimedia streams across the networks are introduced.

2.4.1.1 Internet Protocol

IP is the common network layer protocol for the Internet, and is becoming the standard network layer protocol for packet based 3G wireless networks and beyond. IP is responsible for providing addressing and routing/forwarding, fragmentation/reassembly, and acts as the common interface between the applications. Currently there are two versions of this protocol in use: IPv4 [BoulW] and IPv6 [DeerS]. IP packets, by default, can only provide a best-effort forwarding service. However, if the network routers are programmed to support different levels of services, then the type-of-service (ToS) field on the packet header can be used to indicate any service requirement.

2.4.1.2 User Datagram Protocol

The User Datagram Protocol (UDP) is a transport layer protocol that allows the transmission of IP packets without having to establish a connection [PostJ]. UDP is basically an interface between IP and upper-layer processes. UDP protocol ports are able to distinguish multiple applications running on a single terminal from one another. UDP is an unreliable protocol and does not provide functionalities like flow-control or error-recovery for IP the layer. However, UDP is simple contain fewer bytes and consume less network overhead than TCP. UDP is useful in situations where the reliability mechanisms of TCP are not necessary. Real-time multimedia applications are transported using UDP since it can provide a fast transmission of media packets without needing to wait for any acknowledgement signal.

2.4.1.3 Real-Time Transport Protocol

Real-Time Transport Protocol (RTP) is an end-to-end protocol for the transport of real-time data (i.e., multimedia streams) [SchuH]. RTP provides mechanisms for synchronization, framing, encryption, timing, and source identification of multimedia streams. RTP is usually used together with RTP control protocol (RTCP), which provides periodic report regarding to QoS received at the destination end systems. RTP is typically used directly on top of UDP/IP. RTP has the unique ability to carry information about the application type it is carrying. This enables the intermediate
systems to recognise the packet contents and if necessary take an appropriate action. In addition to the base RTP specification, a number of companion documents exists that provide encapsulations for various media formats including iLBC, MPEG-4 etc.

### 2.4.1.4 Transport Control Protocol

The TCP provides reliable and connection-oriented transmission of data in an IP environment. TCP is a transport layer protocol and is used on top of IP. TCP has numerous functionalities including stream data transfer, reliability, efficient flow control, full-duplex operation, and multiplexing. It packetises the user data into segments with a forwarding acknowledgment number that indicates to the destination the next byte the source expects to receive. By this way, TCP allows systems to deal with lost, delayed, duplicate, or misread packets. Due to its acknowledgement based operation, TCP cannot provide support for multimedia applications. In addition, TCP is also not suitable for multimedia applications since they are not tolerant packet level flow control. However, a TCP-friendly operation is important for multimedia applications and congestion prevention in IP networks [FloyS2].

### 2.4.2 QoS Architectures for IP-Networks

The default service offering associated with the IP-networks is the best-effort service model which makes no attempt to differentiate between different media streams generated by various users. In addition, no guarantees regarding the availability of network resources or any means of grade of service is offered. This scheme works fine when the network traffic is flowing smoothly and there is no congestion in any part of the links between the network nodes. Unfortunately, the IP network traffic is not very predictable and can be bursty at times. That is to say, in the case of network congestion, data packets are either delayed or dropped. There have been attempts to augment this base service with a number of selectable services, which in turn would provide better services during the congestion times. The Integrated Services (IntServ) model [BradR] and the Differentiated Services (DiffServ) model [BlakS] were developed to serve such purposes.

In the Intserv approach, the application first signals its service requirements to the network, and the network responds this by indicating whether the network load is able to carry the requested traffic load. In certain cases, this request might be rejected straightaway during the admission request signalling if the user is not eligible to make any requests about the traffic it is intending to transmit. If the network indicates that it cannot accommodate the requested load, the application can instead use the best-effort delivery scheme. The process of service request takes place within the Resource Reservation Protocol (RSVP) and the end-point demands are passed on to the network. If the network decides to grant the request, the required network resources (i.e., the
bandwidth and computational resources) are reserved on the end-to-end path for the application. The resources are kept reserved unless one of the end-points tears up the connection or the network signals that it can no longer accommodate the reservation.

The IntServ model imposes per application state within the network and is not viable for large-scale networks (considering the heterogeneity of the network end-points and older sections of a network). Nevertheless, it can be useful for small-scale networks such as LANs. Resource reservation is expensive and is also not feasible if scalability is an issue of concern.

In the DiffServ approach, the application is decoupled from the network load function. The traffic control functionality is implemented as a part of the function of admission of traffic into the network, admitting a predetermined load of traffic within each service category. The application makes no negotiation about the service required but instead marks the packets with a code to indicate the desired service. On the entry to the network, each packet from different users is classified into a particular service profile. Depending on the service class of each packet, the network offers a different type of treatment.

The DiffServ approach allows a better management of network resources and integrity of networks services together with superior scaling properties. It is not expensive to implement and does not require the network to install per-flow state information in every network node along the end-to-end path. However, it does not incorporate any concept of control signalling to inform the traffic conditioning elements of the current state of the network, or the current per-application requirements [HustG]. In other words, there is a level of imprecision in attempting to match applications' service requirements to the network's service capabilities.

Neither IntServ nor DiffServ models have become widely implemented approaches for QoS provisioning for IP based networks. One reason for this is that neither of the models was designed to cope with the special needs of particular applications and they are only concerned with the delivery of the data only. That is to say, the service needs are assumed to be mainly transport and network layer issues. There have been some research efforts to integrate these two models in order to build a more flexible architecture, which can complement the disadvantages of both models [BernY]. Unfortunately, even this approach seems to be insufficient to support widespread use of QoS-based services on large-scale IP networks such as the public Internet [HustG].

Nevertheless, DiffServ, with its scalable nature, is a highly favourable approach in terms of providing transport level QoS for different user applications. For this reason, Chapter 4 of this
thesis introduces the possible advantages of utilising the DiffServ model for multimedia applications.

With the ever growing interest on the multimedia applications and the tremendous growth in the Internet and wireless communication technologies, it is certain that some form of traffic differentiation is essential for providing basic QoS (better than best-effort) in the IP-networks. However, in the view of the author, advanced QoS provisioning will only be possible with the introduction of intelligent application-layer services, and consequently provision of per-application based content adaptation techniques in the networks.

2.4.3 QoS for Wireless Multimedia

Multimedia services over a wireless channel are subject to high bit-error-rates due to channel fading, noise or interference, and intermittent connectivity due to handoff. What is more, the bit-error rate in the wireless channel is time varying and has a difficult to predict characteristic. On the other hand, the end-to-end delay in the network is usually higher than the fixed network case, particularly for wireless packet networks. The major reasons for that are transcoding for transfer over the radio interface, adding error correction information and interleaving over a number of radio frames [MarcB].

First and second generation cellular technologies were tailored to support voice and low speed data services. However, third generation mobile systems, such as Universal Mobile Telecommunication Service (UMTS), are being designed to provide multimedia services in both circuit-switched and packet-switched fashion. This research only concentrates on the packet-switched architecture of UMTS.

UMTS network uses Wideband Code Division Multiple Access (W-CDMA) [TaciK] as its Radio Access Network (RAN) technology and theoretically is able to support bit-rates of 144kb/s in rural, 384kb/s in urban, and 2048kb/s in indoor environments [TS22.105]. However, typical bandwidth that will be supported by W-CDMA for video applications is in the range of 64-384kb/s [EtohM].

UMTS is able to support end-to-end QoS guarantees for its packet domain, through the interaction of bearer services established between its modules at different layers. The UMTS bearer services support not only the specification of priority and bandwidth requirement type of parameters, but provide for a rich set of other service quality attributes. The QoS requirements of each traffic class are defined in terms of UMTS bearer service attributes. These can be categorised into radio access
Chapter 2. Multimedia Communications

and core network bearer service attributes. The attributes are defined in a way that the system is not tied to any particular transport architecture, hence allowing flexibility for the network to evolve as the transport technology advances. The list of these attributes, their values and applicability to each traffic class is given in [TS23.107].

Since UMTS networks are committed to provide multimedia services ranging from conversational media to non-real-time communications, there are four distinct QoS classes defined within the design of the architecture. It is claimed that from the end user point of view, the impression of the connection quality is mainly related to the delay experienced on the connection [KaarH]. Hence, the connection delay is the main separating attribute between the UMTS QoS classes. There are other factors, which are also taken into account, such as bandwidth and the nature of the traffic (symmetric/asymmetric). The UMTS QoS classes are as follows:

- **Conversational Class**: minimum fixed delay, no buffering, symmetric traffic, guaranteed bit-rate.
- **Streaming Class**: minimum variable delay, buffering is allowed, asymmetric traffic, and guaranteed bit-rate.
- **Interactive Class**: moderate variable bit rate, buffering is allowed, asymmetric traffic, and no guaranteed bit rate.
- **Background Class**: large variable delay, buffering is allowed, asymmetric traffic, no guaranteed bit rate.

The conversational class represents the most demanding QoS class where conversational type real-time traffic is of concern. The streaming class does not set tight limits for the delay and allows some buffering in the network. The streaming class services are typically one-way, presentational type of applications, e.g. user listening radio through his mobile handset. The interactive and the background classes are used for services that are not very sensitive to end-to-end delays. Under these QoS classes, the delay between request and response may vary and the information to be delivered to the user can be buffered in order to optimize the network performance and capacity.

If a UMTS network employs IP transportation mechanism or Asynchronous Transfer Mode (ATM) transport, the UMTS service classes should be mapped onto the QoS services offered by IP or ATM, i.e. packet classification schemes as in the DiffServ scenario can be used. This issue is further elaborated in Chapter 4.
Chapter 2. Multimedia Communications

It is apparent that even the UMTS architecture has used a rather approximated method in determining the requirements of end-users. That is to say, for real-time applications, UMTS mainly assumes that if a certain delay bound and bandwidth is provided, user satisfaction can be obtained. The UMTS QoS parameters mostly apply to lower protocol layers, and are not meant to be directly observable by the application. Based on these observations, it can be claimed that that UMTS can also benefit from certain QoS adaptation techniques where multimedia streams are tailored to specific user requirements and instantaneous channel conditions.

2.5 QoS Adaptation Techniques

The success of the Internet and the second generation mobile systems has motivated the development of various enhanced-capacity fixed and wireless networking technologies (e.g. 3G, WLAN, Broadband Internet etc.). The services supported by such networks helped the emergence of a new vision of being connected at anywhere, anytime and with any device. However, this has created a wide range of heterogeneous systems, which has resulted in the lack of efficient communications between multimedia users lying in different network domains.

Ideally, networks should be able to provide seamless QoS support for multimedia applications by being able to adapt to the transmission channel problems and diversities of the heterogeneous systems. In practice, it may not be feasible to create a universal framework that would solve all the problems associated with the heterogeneity. Although IP is the common communication medium for different packet networks, diversities at various levels makes it difficult to maintain effective end-to-end QoS performance for the multimedia applications.

At the network level, diversity exists due to the differences in network size, latency, capacity, link bandwidth and QoS policies used. This requires bilateral agreements between different network operators which specify how the cross-network traffic should be handled. In other words, network operators need to establish admission control, resource allocation, traffic aggregation, congestion control, and prioritised forwarding policies for the traffic flows of different applications.

At the application level, different multimedia applications have different QoS requirements in terms of packet delay, loss, jitter, and error resilience. On the other hand, users of a network may have different terminal capabilities, perceptual requirements, service-level agreements, and they may experience different channel conditions depending on the network access technology used. Tackling these diversities call for effective adaptation mechanisms which can tailor the network behaviour to the application/user requirements and the applications to the network conditions.
Existing adaptation schemes can generally be classified as end-point (sender/receiver) and network node.

Considering that the focus of this thesis is on video-based applications, the following subsections introduce the related research about different QoS adaptation techniques and discuss their pros and cons.

2.5.1 End-Point Based Adaptation

In the end-point based adaptation mechanisms, users or content servers are responsible for providing the adaptation to the user requirements and changing channel conditions. End-point based adaptation has the advantage of not adding on extra complexity on the network. In this approach, users make use of end-to-end signalling to communicate their capabilities and requirements. When network adversities arise, end-points need to signal their condition back to the source and expect the necessary adaptation to take place. In general, the adaptation is realised either by using the functionalities of the application, or execution of an algorithm/protocol which can adapt the application to the network conditions.

The basic adaptation technique to network congestion is to encode multiple versions of the same video content and switch between different streams based on experienced network conditions. This is also called the simulcast technique [CheuS]. It is a simple approach and has been adopted in many commercial video streaming products.

Another form of adaptation can be performed by controlling the encoding parameters based on the monitored congestion levels in the network. A number of such approaches have been published in the literature [WuD] [RejaR]. However, these approaches try to model the video rate adaptation in way analogous to TCP flow control mechanism, which results in high fluctuations in the output video quality and perceptual discomfort.

The use of layered video coding schemes presents an alternative solution for QoS adaptation. In layered coding, rate adaptation can be performed by adding and dropping layers according to the network conditions experienced by the end user. McCanne et al. [MaccS] proposed the receiver driven layered multicast protocol where each video layer is transmitted over a different video channel. The packet-loss levels experienced by the users are used to decide on the channel a user should join in order to receive an optimised video performance. Another interesting work was presented in [VickB], where the bit-rate assigned to each video layer was adjusted dynamically.
based on the congestion feedback obtained from the network. The disadvantage of utilising the layered approach in video coding is its complexity and coding efficiency.

Providing error-resilience for video applications is another widely adopted adaptation technique. Commonly, FEC-based mechanisms [BoloJ2], [WuD], [CaiJ], link adaptation [KodiC3], and error-resilient encoding are used to provide robust video communications [GiroB].

There are a number of common drawbacks associated with these approaches. In summary, they lack in scalability, make inefficient use of network resources, and cannot account for the requirements of different users, their terminal capabilities, and the differences in applications in a heterogeneous networking environment.

### 2.5.2 Network Node Based Adaptation

Network node based adaptation strategies have been proposed to alleviate some of the disadvantages associated with end-point based approaches. Specialised network nodes (proxies/gateways) are strategically placed in certain locations of the networks to alleviate the adverse effects of system heterogeneity. The operation of such nodes is usually defined by transcoding, filtering, and caching mechanisms. Network node based adaptation strategies have the advantage of offloading the multimedia servers, and provide fast and dynamic response to changing network conditions and user requirements. Such mechanisms are usually pre-configured to perform a specific action on the input multimedia content, although limited configurability is also possible (e.g., changing the adaptation parameters based on measured/estimated channel conditions). In addition, network node based adaptation can also involve interactions with end-users to determine a suitable adaptation strategy.

Filtering is the process of dropping the right packets in order to preserve the perceived multimedia quality. There are various types of network based adaptation mechanisms which fall in the filtering category. Dropping the enhancement layers or frames (B-frames or P-frames) [HeD] in the network are the commonly used filtering methods.

Caching mechanisms are deployed in caching proxies, which are positioned at the network edges close to the users. The responsibility of these proxies is to store the most popular multimedia content between the servers and the users. By this way, a reduced response time (i.e., end-to-end delay) to their requested content is experienced. The other advantage of caching mechanism is that the possibility of experiencing packet losses are reduced since there are less number of links that should be traversed between the proxies and the users. Caching also reduces the network load.
by effectively reducing the number of sessions between the servers and the clients. However, in an environment where users access the network using a variety of devices with different capabilities (i.e., pervasive computing environment), the efficiency of caching strategies is greatly reduced. This is because the server is unable to produce multiple versions of the same content or the proxy has a limited storage capacity or both. Therefore, a different caching strategy can be followed which only stores the high quality version of a content and uses transcoding to adapt the content to the requirements of users [ShenB].

Transcoding is the most prominent technique used for video adaptation. Transcoding architectures are usually employed in the network nodes to adapt certain coding characteristics of video streams to heterogeneous systems and user requirements. There are numerous applications of transcoding. Commonly, they are used for bit-rate regulation, format conversion, resolution scaling and error-resilience.

The objective of bit-rate reduction is to adapt the compressed video stream to the changing channel conditions and bandwidth requirements of different users. Applications requiring this type of conversion include television broadcast, Internet video streaming, and video-based applications over the wireless networks. In [KumaR], a transcoding mechanism which regulates the video bit-rate based on the network load was proposed. A more comprehensive approach was presented in [DogaS] where a video gateway was used to provide rate adaptation through rate-control and frame-rate transcoding. On the other hand, [BjorN] discussed the implementation techniques used for designing a rate-reduction transcoding algorithm that is targeted for non-scalable coders. A low-complexity transcoding architecture was proposed in [VetrA2]. This transcoder is targeted for wireless video streaming applications and able to perform rate-control through requantisation and selective frame dropping algorithms.

Application incompatibilities amongst network end-points call for format transcoding mechanisms to be in place. Basically, format transcoding involves the transformation of one video coding standard into the other [DogaS2], [FeamN], [NakaY].

When the users are accessing the network with various different terminals which have different screen sizes and processing capabilities, downscaling of the compressed video into lower spatial resolutions becomes a necessity. Examples to this kind of adaptation are given in [ZhijL] and [ShanT].

Transmitting video over error-prone channels (e.g., wireless networks or congested links) requires taking into account the conditions in which the video will be transmitted. Adaptation to hostile
channel conditions can be performed using error-resilience transcoding mechanisms. Such mechanisms have been proposed in [DogaS3], [EminS] and [ReyeG]. Error-resilient transcoding approach together with a bit-rate adaptation mechanism has also been investigated as a part of the research work presented in this thesis. Details of this study are introduced in Chapter 5.

In summary, network-based QoS adaptation is an effective way of overcoming the difficulties imposed by the system heterogeneity and different user requirements. However, most of the traditional network-based solutions involve placing of a specialised gateway/proxy node at a specific location in the network. In this approach, these nodes require frequent software and hardware upgrades as the network configuration, number of users, and the characteristics of applications change over the time. In other words, although this adaptation approach has been proven to be useful, the scalability of scalability of the provided services is an important concern.

Ideally, end-point and network node based applications should be utilised together in order to maximise the application quality perceived by the end-users. In that sense, it is unfair to make a comparison between these two approaches. Nevertheless, in an environment where end-users have insufficient resources to perform the necessary adaptations, network node based approach can be regarded as the more effective solution. The challenge in this approach is how to implement the adaptation mechanisms such that they can be activated at the right location and the right time, as well as being scalable to the changing user and network requirements.

2.6 Evolution of Applications and Networks

Users are accessing their networks with a variety of different devices that show enormous divergence in their capabilities in terms of processing power, display and speaker characteristics, and human-interface modules. Thus, the multimedia content is required to be tailored based on the user requirements, context, and the terminal capabilities at anywhere and anytime. Moreover, advances in the computing and audiovisual device technologies are enabling the development of rich multimedia applications (e.g., tele-presence, virtual reality, 3D audiovisual content etc.). These applications are sophisticated in their presentational characteristics and demand specialised treatment from the network. That is to say, defining the requirements of such applications with a number of network parameters (e.g., delay, bandwidth, packet-loss etc.) would not provide the desired levels of QoS. Therefore, it is essential to provide a well defined and functional network support for the current and future multimedia applications.
In the earlier sections of this chapter, the general characteristics of multimedia applications and their QoS requirements were introduced. In addition, the existing networking approaches for providing better than best-effort forwarding service were outlined. The amount of research performed on the QoS adaptation techniques shows that the existing IP QoS architectures are not sufficiently effective in satisfying the diverse requirements of different multimedia applications. Although it is undeniable that such architectures are useful, their lack of flexibility and limited processing ability brings in the need for advanced networking architectures which can enhance the range of services offered to multimedia applications.

IP QoS architectures that are able to provide limited processing of user traffic can be seen as the first step in the evolution of IP networks from their best-effort nature to an application centric and more functional structure. Introduction of media proxies and gateways is the further step in the evolution process. Such devices can provide content adaptation services which could initially performed only at the network end-points. Content adaptation operations, like described in Section 2.5, are a series of operations that provide the processing of multimedia content into the required form or mode.

In the last few years, there have been considerable industrial efforts devoted to the research and development of Content Delivery Networks (CDN) and Content Services Networks (CSN) [MaW]. These are overlay networks deployed at the network edges to provide application specific services for the users. The overlay network structure contains proxies/gateways in the form of dedicated and custom networking devices that are responsible for performing a series of pre-specified and fixed tasks on the application traffic. The aim of introducing such functionality in the network is to provide value-added services that can meet the increasing requirements of users and applications, and lead the way towards the next generation Internet. The disadvantage of this approach is its lack of flexibility. When a new service need arise, the service provider has to upgrade its network nodes, and the deployment of new services can be subject to standardisation. In addition, the users' involvement in the service creation process is extremely limited. Nevertheless, the CDN/CSN initiative stands as a courageous step forward in introducing large scale computation and storage inside the network for the purpose of providing application specific services to the end users. The success of such architectures would encourage the adoption of more sophisticated but flexible networking approaches which can provide user specified computations to take place inside the networks.

Active (or Programmable) networking [TennD2], [CampA] is an alternative and innovative approach to the traditional packet networking architectures. It represents a significant step in the evolution of packet networks, from traditional packet-forwarding engines to more general
functionality supporting dynamic control and modification of network behaviour [CalvK]. The telephone networks started with switches, which embodied most of the intelligence of the networks and terminals, with no intelligence. The Internet model on the other hand had smart terminals and non-intelligent routers. However, in the active networking architecture, both terminals and the network nodes are intelligent, enabling the ultimate flexibility and service provision.

Active networks enable customisation of the network nodes so that application-specific services can be applied to user packets. The users can perform this either by uploading software components into the network nodes or invoking/combining the existing ones that were installed by the network operator. Invoking or combining service libraries, or uploading service components can be performed by injecting user programs into the intermediate nodes of the network. As a result, the active nodes can identify/classify the user-data and provide necessary processing required for enhancing network communications. In short, active networks support dynamic modification of the network's behaviour as seen by the user.

With the radical changes brought by active networks notion, the future networks can be envisaged to become a virtual extension of the user's terminal. By utilising the computing resources in the networks, user terminals will be able to receive enriched multimedia applications and services, which would not be otherwise possible due to the limitations imposed by their hardware and bandwidth profiles. In addition, the user will be able to modify the network behaviour based on his/her requirements. However, this does not mean that the networks will be completely open to user modifications. The level at which users will be allowed to create, deploy or initiate network services should be determined by the service providers. The issues regarding to the management of such networks are beyond the scope of the research work presented in this thesis.

Performing user defined/selected computations would obviously demand a large amount of processing power and memory resources to reside in the networks. However, the continuous advances in the VLSI technologies are the motivation behind such initiatives (i.e., programmable networks). For example, the state of the art desktop personal computer microprocessors, like Intel® Pentium® 4, are running at clock speeds above 3GHz, and by year 2010 this is expected to increase to 15GHz. Such rapid advancement in the hardware capabilities is an indication that cheap and real-time multimedia computations in the networks is not far from realisation.

In the view of the research work presented in this thesis, the increasing complexity of multimedia applications and the respective user requirements will be provided with advanced QoS support through the use of active networking technologies. Therefore in this thesis, the investigation of
how multimedia applications and users would benefit from application specific services deployed in the networks is made. These services refer to network-based computations that are performed on the multimedia applications to improve the QoS experienced by the users.

### 2.7 Concluding Remarks and Recommendations

In this chapter, an overview of different aspects of multimedia communications has been introduced. In particular, the concept of QoS provision has been elaborated from the network, application, and user point-of-views.

From the networking point of view, delay, jitter, packet loss, loss pattern, and the bandwidth are the parameters that determine the QoS received by the applications. Applications have specific requirements regarding to these parameters. Each parameter has a different effect on the application performance as described in Section 2.2. On the other hand, users only experience the cumulative effect of the adversities associated with such parameter and are not in a position to specify any specific requirements in this sense. From the user point of view, the perception of media quality is also dependent on a number of psychological factors associated with the application response time, predictability, and consistency.

In the design of an effective QoS architecture, different requirements of applications, users, and the capabilities of the network infrastructure should all be considered in conjunction with each other. That is to say, applications should be tailored to the networks and the networks should provide the necessary adaptation tools for applications. For this purpose, the multimedia applications should be classified according to their characteristics. Based on such classifications, network services can be built to meet the demands of the applications. In addition, application developers can utilise such information to develop robust and flexible applications that can be adapted to network adversities. In Section 2.3, multimedia applications were coarsely classified into two broad categories as voice and video based applications. This is because the concentration of this thesis is on low bit-rate multimedia applications (i.e., particularly video-based applications). A discussion of the effects of different network parameters on the perceived media quality has also been introduced. Particular emphasis was made on the effects of packet loss and bit-errors on the video quality since the later chapters of this thesis make assessments of video performance with respect to these parameters.

In Section 2.4, multimedia transport issues were discussed. The IP protocol is seen as the common interface that enables interoperability between different networks. Due to massive deployment
and adoption of IP based communication infrastructures, it is unlikely in near future to witness the emergence of an alternative protocol that would revolutionise the way packet-switched networks operate today. Therefore, the efforts on building an effective QoS framework should be within the scope of IP based communications. In that respect, a brief summary of the existing IP-QoS architectures were introduced. QoS provisioning in wireless networks is on the other hand somewhat different issue. IP-QoS architectures have also been considered for the wireless networks (i.e., UMTS). However, UMTS enables a more elaborate definition and provision of application QoS through the interaction of bearer services established between its modules at different layers. Supported applications are categorised into a number of discrete classes and the QoS requirements of each traffic class are defined in terms of UMTS bearer service attributes.

Although IP QoS architectures are known to be useful in providing better than best-effort support for the applications, diversities at various levels makes it difficult to maintain effective end-to-end QoS performance for the multimedia applications. Therefore, application developers and service providers are relying more and more on QoS adaptation mechanisms that can be introduced at the end-points and intermediate network nodes. A number of adaptation mechanisms have been introduced in Section 2.5 with a particular emphasis on network based adaptation mechanisms such as transcoding proxies/gateways. There are inherent problems regarding to placement of such static value-added service points in the network as highlighted at the end of Section 2.5.2. The main problems are related with scalability and the flexibility.

In their current states, IP networks do not incorporate the necessary tools that could provide an effective end-to-end QoS for multimedia applications. Therefore, networks are required to evolve from their closed system form to an open, programmable, and adaptable form that can suitably adapt the applications to the requirements of the users as well as the networks. In effect, networks should move from the concept of being a data delivery medium to a service delivery medium. In this thesis, it is envisaged that this evolution will be through the path of active networking concept which is the subject of the next chapter.
Chapter 3

3 Active Networks

3.1 Overview

Today's packet-switched networks, which make use of the store-and-forward model, are expected to evolve into a service centric medium where the network services will be customised to the user and application requirements. In this research, it is hypothesised that this evolution will take place in lines with the active networking concept. At the onset of this evolution, the active networking components are expected to be deployed gradually and coexist with the packet forwarding entities of today's traditional packet switched networks (passive networks). Later, open and programmable interfaces with varying user programmability will spread ubiquitously, modulating the nature of networks from packet forwarding engines to service delivery media.

The aim of this chapter is to introduce the background knowledge about the active networks and describe how this approach may lead the way towards the ultimate service provision flexibility in packet-switched networks. In the following sections, the fundamentals of the active networking concept, various approaches in active programmability, applications of active networks, and the challenges being faced are introduced. The implications on multimedia communications and QoS provisioning as well as the related research are discussed as a basis for further discussions throughout the thesis. Overall, this chapter establishes the foundations for the active services which are presented in Chapter 5.

3.2 Background

The success of passive networks (i.e., the Internet) is based on its design principles defined by the end-to-end arguments [SaltJ]. These principles are about the organisation and placement of functionalities in networks. In essence, end-to-end arguments state that many functions can only be completely implemented at the end points of the network and intermediate nodes should remain as simple store-and-forward engines. Therefore any attempt to build computationally complex features in the network to support particular applications is considered to be unpractical unless they are implemented at the end-points. In [SaltJ], it is argued that implementation of such
complex functions is rarely necessary, and that systems designers should avoid building them into the networks. In the light of these arguments, the passive networks are designed with simple router architectures, which perform limited processing inside the network. This processing is mainly in the form of physical, data link and network layer operations. The idea behind this is to ease the processing load on the intermediate network nodes and push it to the end-points to speed up the packet-forwarding process. The purpose of passive networks is to deliver the data from the source point to the destination in a timely and robust manner rather than providing specialised services for supporting the application and user requirements. In that respect, the router would not attempt to access or alter the packet beyond its network-layer header. The end-to-end operation of passive networks with respect to seven-layer International Organization for Standardization (ISO) protocol stack is shown in Figure 3.1.

![Figure 3.1: Operation of Passive Networks](image)

Any information regarding the packet is embedded in the header data. Routers use this information as the input to the router functions that process the packet i.e., routing, queueing, scheduling algorithms. In its best-effort nature, all packets get equal treatment regardless of their content and source. Traffic flows are subject to services designed and implemented by the network operator. Because of this predefined and inflexible way of handling traffic flows, the requirement and consumption of processing resources in each network node is well known in advance.

The service requirements of applications and users are exceeding the capabilities of the traditional passive networks. Passive packet networks have the drawback that the intermediate nodes are closed systems whose functions are rigidly built into the embedded software and their hardware. Therefore, these networks operate in a predetermined and fixed fashion where capabilities are restricted by the standardized protocols installed throughout the network. Standardization of
protocols is a long process, which may limit the range of services that can be provided in a system. In the era of rapidly evolving technology, diverse range of applications require various different services and protocols to be deployed dynamically so as to be able to satisfy network, user, and application requirements.

Recent advances in the distributed systems and transportable software along with increasing application demands for better control of network resources have motivated the research for alternative networking approaches. In addition, it has become apparent that there is a convergence between computing and telecommunications technologies, which resulted in the proliferation of the demand for various multimedia applications. Therefore, there is a need to reconcile the perspectives of the computing and the telecommunication communities in new dynamically programmable network architectures that can support fast service creation, resource and QoS management through a combination of network aware applications and application aware networks.

3.2.1 What is Active Networking?

Active networking is a new networking paradigm where the network is transformed from a simple store-and-forward engine into a dynamically programmable and fully flexible mechanism, which can offer intelligent services both at control and data planes. The word *active* indicates that nodes can perform computations on the user packets and modify their contents. In addition, this processing can be customized on per user or per application basis [TennD]. About a decade ago, the initial active networking concept emerged from the discussions within the Defense Advanced Research Projects Agency (DARPA) research community. These discussions were aimed at identifying the future of networking systems by considering problems of the passive network systems. Several strategies, collectively referred to as active networking, were devised to overcome the problems associated with networking systems. Later, this concept has attained widespread interest amongst the networking community. From 1996 to 2002, DARPA has funded over 50 projects and motivated the establishment of a testbed called ABONE [ABONE]. Active networking concept has also attracted the interest of numerous researchers in Europe, South East Asia and the rest of the world. In 1999, an international conference called IWAN (International Working Conference on Active Networks) was put forward against the background mentioned earlier, and six annual conferences were held in different countries. Other international conferences and workshops, i.e., OPENSIG, ANTA and DANCE, have also been dedicated for active networking. In Europe, active networking research was mainly carried out in the context of the FAIN project [FAIN].
The reason why active networks attracted so much attention is because it is thought to represent the evolution of packet networks, from traditional packet-forwarding engines to more general functionality that supports dynamic control and modification of network behaviour [CalvK]. Active networks enable customisation of the network nodes by exposing the underlying nodes’ resources and mechanisms through a programmable interface. By this way, application-specific services can be applied to traffic flows. The users (e.g., network administrators, service providers and customers) can program the network either by uploading software components into the network nodes or invoking/combining the existing libraries that were pre-installed by the network operator. The programming of the intermediate nodes is performed by the execution of the methods carried in active packets. As a result, the active nodes (e.g., routers, servers, proxies etc.) can recognise, differentiate, and process the user traffic to provide the necessary service support for improved end-to-end communications. When considered in the context of layered OSI model, active nodes are network entities which can, depending on the service required, extend their processing all the way up to application layer. Figure 3.2 depicts the operation of active nodes in the context of OSI model.

3.2.2 Advantages of Active Networks

Active networks enable a highly flexible and extensible network infrastructure, which can support dynamic modification of the network’s behaviour as seen by the user. Such flexibility and dynamic control has advantages on multiple levels.

- From the network operator’s point of view, active networks have the potential to reduce the time required to develop, test and deploy new network services. In today’s
networks, due to economical reasons, the shared infrastructure evolves at a much slower rate than the computing technology. A major consequence of being able to change the behaviour of network nodes when and where necessary is that it would be possible to modify network functionalities and management structure with far less changes on the network router/switch hardware (i.e., compared to the amount of effort required to introduce a new service in traditional networks) without going through a lengthy standardisation process.

• For service providers, the primary benefit of active networking would be the added flexibility that enables them to react quickly to changing network and customer requirements. Like network operators, service providers would also benefit from reduced time-to-deploy of new services, better network management capabilities and more control over network resources. New charging schemes would be available, which could allow charging based on the service that customers may choose according to their requirements.

• Active networking would allow third-party network software developers to compete with established network vendors and sell service software in much the same way as for end systems today. This would create a new competitive market for network software, services and applications.

• From the end-user point of view, active networks can enable users to choose from numerous available network services, and request and tailor them to their particular applications and interact with the network to indicate their QoS and presentational preferences. This can be done by choosing the services required that might be mapped onto the network parameters (customised scheduling, queueing etc.) in the form of user-code, and to program the dedicated active routers to apply high level services to their applications (e.g., spatial scaling of a video application through transcoding services). Users could be charged according to the service requirements they choose.

• Finally, for researchers, a dynamically programmable network offers a platform for experimenting with new network services and features on a realistic scale without disturbing regular network services [CalvK].

Users are expected to play a crucial role in the development and deployment of active networking technologies. Their demand for advanced network services would be the impetus in the involvement of numerous service and application developers in the development process. Moreover, the market pressure resulting from the rising user demands might be the catalyst that can make network operators, service providers, network software developers and equipment vendors move to a network architecture which provides the desired flexibility.
3.2.3 Active Networking and End-to-End Arguments

As motioned earlier, end-to-end arguments provide the guidance for placing functions within a computer network. The success of the Internet is based on designing its networks and applications in line with these principles. According to these arguments, implementation of any high level functionality within the network is precluded. Otherwise the network would change from a generic transmission medium to a complex architecture whose services are optimised for only a particular set of applications. Therefore, a function or service should be carried out in any networking layer only if it is needed by all clients of that layer, and it can be completely implemented in that layer. This means that a communications layer (as in OSI model) should not be designed to offer services, which only other layers can implement completely, unless it is a clear performance enhancement for the whole system.

Active networking is about installing programmable services for user applications in the network, which previously could only be realised at the end-points. Thus, at first glance, it appears to contradict with the end-to-end principle. However, programmability may allow a network client to implement precisely the service it needs; a property in agreement with end-to-end arguments. Therefore, applying end-to-end arguments should be carefully reviewed and not applied in a definitive way. Instead, the specifics of each particular active networking idea would benefit from evaluation in light of the end-to-end principle [ReedD]. In [BhatS], it is claimed that active networking is a natural extension of design principles prescribed by the end-to-end arguments.

Many applications can be supported better by utilising the information that is only available inside the network. Similarly, network performance can benefit from certain information available at the application/user end. Therefore, it is desirable to combine the network and application information in order to optimise their performance. In this respect, active networks have the potential of enabling application aware networks and network aware application through its programmable interface. Hence, it could be argued that active networking is a consequence of the end-to-end arguments. Nevertheless, support for programmable services in the network may incur considerable costs that may affect most of the users, whether they take the advantage of using such services or not. However, the end-to-end arguments necessitates that if a new functionality is to be implemented in a networking layer, all applications should benefit from it. Unless all applications will make use of such service, then these applications should not be made pay for the cost of supporting it in the network. Thus, in order to support high level functionality in the network, the interface to such functionality should carefully be designed by considering the network wide cost and benefits of such support. The cost of using the interface should be determined based on the application used as this is dictated by the arguments. For example, an
application that is content with the best-effort datagram delivery should not suffer because of the increased flexibility offered to other applications. The key challenge in active network design is therefore to identify the cases in which performance gains and enhanced capabilities can justify the cost incurred in using the active networking architecture.

To assess the aforementioned trade-offs and examine the suitability of deploying a programmable service to a specific network layer in terms of end-to-end arguments, [PartC] has proposed a simple model composed of a set of questions.

- Does adding a programmable service at layer $N$ enhance performance at layers above $N$?
- Does adding a programmable service at layer $N$ allow the elimination of support at layers above $N$?

After a series of analysis, [PartC] concludes that the network can satisfy the end-to-end argument requirements and enhance performance by exploiting programmability at all layers except the networks layer, since network layer provides the interconnectivity and communication amongst arbitrary number of heterogeneous devices. On the other hand, programmability at the application layer is the one that is most likely to be beneficial in terms of improving the network and the application performance.

In conclusion, it can be argued that active networking, if implemented appropriately, should not contradict with the end-to-end arguments. Yet, costs and engineering trade-offs associated with deploying programmable services in a specific network level should carefully be evaluated. It should also be noted that there are views within the research community regarding to whether end-to-end arguments are still valid for future's advanced applications and user requirements. Some researches would like to consider the future's network as a virtual extension of the user's equipment and claim that new system design principles may be required for realising the full benefits of the advancing technologies. In fact, the scientists who created the end-to-end arguments had not foreseen the success of the Internet. However, many believed that this success is based on these design principles even though the Internet was never meant to become what it is today. In the same way, it can only be predicted but not foreseen what the future networks will be like. Therefore, it is scientifically reasonable to question the applicability of end-to-end arguments and seek for alternative approaches for future networking systems.
3.3 Active Network Architecture

The active networking research had mainly flourished under the DARPA funded programme [DARPA]. Under this programme, an informal working group was established to define the scope and principles of active networking along with an architectural framework [CalvK2]. However, there exist a number of active networking approaches which differ from each other in terms of programming interface, programming model, service composition model and security model they utilise [SchmS]. In spite of the differences, the core concept of active networking and the design principles are same for all approaches. Thus, in the following subsections, the fundamental elements of active networks that enable programmability are described. For this purpose, the architectural framework presented in [CalvK2] is taken as the basic reference.

3.3.1 Design Objectives

The main aim of active networking is to advance today’s networks and applications by enabling user customisability through dynamic service programmability. In the design of an active networking architecture, the following objectives may be pursued [CalvK2]:

- Minimize the amount of global agreement/standardization required, and support dynamic modification of aspects of the network that do not require global agreement.
- Support fast-path processing optimizations in nodes. The architecture should not preclude active nodes from performing standard packet forwarding at speeds comparable to non-active IP routers.
- Support deployment of a base platform that permits on-the-fly experimentation and testing. Backward compatibility with existing network nodes and protocols is desirable.
- Scale to very large global active networks.
- Provide mechanisms to ensure the security and robustness of active nodes individually.
- Support network management at all levels.
- Provide mechanisms to support different levels/qualities/classes of service.

3.3.2 Active Nodes

An active network contains a set of active nodes connected by a variety of network technologies. Therefore, no assumptions are made in the design of an active network regarding the underlying transport technology. The active nodes receive packets from users and other nodes, perform a computation based on the code provided in the packets and forward one or more packets towards the other nodes based on the routing algorithm determined or defined. They perform the functions of receiving, scheduling, executing, monitoring and forwarding packets. The operations of active
nodes are manipulated by the active network packets, which usually are referred to as *active packets*. Although Active Network Encapsulation Protocol (ANEP) [AlexD] is the most commonly exercised active packet format, there is no global agreement in the research community about the format and functionalities of active packets. A detailed discussion to available approaches is given in Section 3.3.3. Basically, the software methods carried inside active packets are executed inside the active node to initiate a service for the user (i.e. user's data traffic).

Each active node runs an operating system called the *Node* Operating System (NodeOS), and one or more Execution Environment (EE). EE implements a virtual machine for interpreting active packets arriving at the node. Different EEs define different virtual machines; one might be completely general and modifiable, while another’s computation simply forwards packets under control of IP header information. Each EE provides an interface through which end-to-end network services are provided to users. In other words, EEs provide the interpretation environment for active services to be initiated through the execution of the code provided within the active packets. All access to network resources is provided through an EE. Another type of EE is called the Management Execution Environment (MEE) which serves as the controlling and monitoring element for the NodeOS and other EEs.

The NodeOS manages the resources of the active node and mediates the demand for those resources, including transmission, computing (CPU) and storage (memory). It also implements communication channels over which EEs send and receive packets [CalvK]. A means of quality of service is provided by NodeOS since its scheduling mechanisms control access to computation and transmission resources of the node. The purpose of these mechanisms is to isolate the user traffic to some extent from the effects of the other users’ traffic so that each appears to have its own virtual machine or virtual link. A typical active node architecture is depicted in Figure 3.3. As seen in the figure, incoming packets are demultiplexed into the NodeOS, where a packet classification mechanism associates data packets with appropriate EEs. The classification mechanism may utilise explicit tags or packet headers (ANEP [AlexD]) to perform this operation. Alternatively, a more generic mechanism that allows classification on arbitrary packet data may be employed. In this case, unlike the explicit method, packets can be transparently classified and assigned to EEs. Hence services can be executed in a transparent manner without involvement of the end-nodes.

End-nodes (user terminals) can also be considered as active nodes with two different modes of operation. In the first mode, the end-user terminal can act as a standard terminal where the application data traffic is produced and/or consumed. This requires the end-user terminals to
utilise a Network Application Programming Interface (NAPI) in order to translate user/application requirements and/or context into program code for active processing inside the network. The second mode is for ad-hoc communications between terminals. In such scenario, a user terminal is required to forward the received traffic to the requesting party. The EE is responsible for implementing the NAPI, while the NodeOS manages access to local node resources by EEs.

![Figure 3.3: Active Node Architecture](image)

Users obtain services from the active network in the form of active services (AS), which are programs determining the operation of the virtual machines residing in the EEs. Depending on the programming model employed by the network, services may be installed by the user on-the-fly or called from an active server on demand. AS can range from data-link to application level services. In this thesis, the focus is on services which are useful in improving the QoS of multimedia applications as perceived by the end-users. A popular example of one such service is transcoding of video streams. The technical insight to these kind services and their performances are given in Chapter 5.

### 3.3.3 Programming Models

Programming models represent the way in which active packets are involved in the execution of the active services in the network. It is the main distinguishing criteria between different active networking approaches. These models can be classified according to the distribution mechanism and the encoding schemes they use for creating the code carried in active packets.
3.3.3.1 Code Distribution Mechanisms

Active packets carry program segments \((active\ code)\) that can be used to build/extent a protocol module on a specific communication layer and/or call a service that is cached inside the active node. Generally, the active networking research community made use of \textit{in-band} and \textit{out-of-band} approaches for code distribution in the active network. Moreover, \textit{hybrid} approaches have also been developed to benefit from the advantages of both. The other method of code distribution is the \textit{code reference} approach.

In the \textit{in-band} approach, each packet contains some code that gets executed at every active node in the network. The network does not store any preconfigured active services; hence the packets are responsible for composing their own services by executing the code they carry at every hop. This approach is considered as the \textit{strong active networking} view and is based on the \textit{active capsules} idea proposed in [TennD2]. Once the code execution is completed in an active node, the allocated resources for active processing are released and the packet is forwarded to the next hop. However, this creates a redundancy in terms of needing to carry the same code for every active node traversed and for every packet. Because of this, the size of the code being carried needs to be limited in order to sustain an efficient use of the network resources. In-band code distribution raises serious concerns in terms of policy, security and network resources. Basically, it is hard to control users from injecting malicious code that would harm the normal operation of the network. Nevertheless, it remains as the most dynamic way of receiving services from the network. This approach is sometimes referred to as the \textit{integrated approach} to active networking since the code is distributed in-band with the data traffic and executed as the individual active packets pass the node. Examples to the integrated approach based designs include; ANTS [ANTS], PLAN [PLAN] and SmartPackets [SMART].

The other approach is the \textit{out-of-band} approach that is analogous to having separate channels for signalling and data transmission in communication networks. This approach is considered as the \textit{moderate active networking} view and sometimes referred to as the \textit{discrete approach} since program distribution is separated from the data transmission. In this approach, all the code is injected into the network before the beginning of the data transmission session. Being able to load active code out-of-band allows for more comprehensive programs to be installed since the required code does not need to be attached with the data packets and can be distributed off-line before any application initiation. In addition, high-level security checks can be performed before a program can be executed on an active node. However, additional latency is inevitable if code is needed to be installed in the network before the session start. Examples to discrete networking approach include NetScript [NETS], Bowman&Canes [BOWC], Joust [JOUST], LARA [LARA], Click [CLICK] and Router Plugins [ROUTP].
There are also hybrid code distribution approaches which are aimed at benefiting from the features of both in-band and out-of-band approaches. One way of providing such functionality would be to enable a discrete active networking approach for the coarse programmability overlaid by integrated approach for fine-grain programmability that enables user defined customisability. The SwitchWare [SWITW] architecture is an example to this kind of active networking.

There exist various variations of the code distribution mechanisms. A more controlled and secure way of providing active network services is realised by the code reference approach. This mechanism can be regarded as a variation of the out-of-band approach, where the code uploading is mainly performed by the service operators. In this approach, the active service modules are cached in active nodes or downloaded from an active service server on demand. The active packets, instead of uploading complete programs on active nodes, carry some program fragments that basically are used to reference a particular service or provide a parameter set that can be used in service composition and customisation. Therefore, the services offered to the customers remain under the control of the service provider. In this scenario, while providing services to the end-users with code distribution approach, the service provider or network operators may make use of out-of-band and/or in-band approaches to dynamically control and manage the network and the services it provides. The operation of the code reference approach in an active network domain is illustrated in Figure 3.4

![Figure 3.4: Hybrid Code Distribution in an Active Network Domain](image-url)
The code reference approach may be regarded as the light active networking view since users have limited capabilities in network programming. In addition, since users are not involved in implementing the services or protocols the need to use, they need to be provided with the information of what services are available in the network. In practice, users cannot decide on where, when and which service to initiate. Therefore, there needs to be a mechanism which translates user and application context and/or requirements into a software code which can be interpreted in the active nodes network. By this way, required active services can be executed with appropriate parameters.

The main disadvantage of this approach is that users cannot compose their own services but only customise the existing ones. The increased network management complexity for controlling the creation, initiation, resource allocation and termination of active network services is the other disadvantage. At first glance, this may seem to have diverted from the original active networking concept [TennD2]. However, it is clear that the exploitation of the potential benefits of active networking concept is difficult to be realised in real life unless the active network functionalities are progressively deployed. The code reference model is the logical step forward in introducing the active networking functionalities within the packet networks. Having been convinced about the usefulness of network programmability through the use of code distribution approach, the service providers and the network operators may be motivated to further advance the system flexibility by introducing other programming models into the network. Therefore in this thesis, it is assumed that the active services are initiated using the code reference approach. In such scenario, the user packets are expected to carry descriptions or hints regarding to the media content type and requirements. During the call setup process, the active network management system acknowledges the user requirements and deploys the required services for the user applications. Active services, on the other hand, use the information embedded on the active header of the application packets to customise a particular service to the user's and application's requirements.

3.3.3.2 Code Encoding

There is various code encoding schemes used amongst the active networks research community. The active network code is represented in the form of a programming language. The choice of the programming language used depends on its properties in terms of mobility, safety, programmability, and execution performance. The existing approaches are:

- Interpreted code.
- Intermediate code.
- Binary code.
- Self-specialising code.
Interpreted code facilitate the implementation of safe execution environments for active code; intermediate code is largely platform independent and at the same time modest in performance; binary programs are preferable for computationally expensive or frequently used modules due to their superior performance; source code based mobile code overcomes the portability problem of binary programs, but at the expense of just-in-time compilation upon code arrival (which limits its usability to discrete approaches). Finally self-specialising code achieves mobility, safety and high performance through dynamic code specialisation at loading time. A brief overview of these code encoding schemes is given in [SchmS].

3.4 Active Processing

*Active processing* (AP) refers to the active service computations performed in an active node. These computations should take place at link speeds, which make active nodes considerably more resource demanding than today's standard network nodes (e.g., routers). Therefore, powerful network node architectures are essential in making active networking viable. Traditional routers can be modified to provide AP either by adding a processing engine at each router port or providing a shared pool of processing engines that can be used to process the traffic from any port. The first method is preferable when all ports have comparable requirements. However, the second approach is better when ports have widely varying needs. Based on the arguments presented in [WolfT], it is possible to design an active router with multiple processing clusters that operate in parallel to provide computational power that is required by AP. The target of such design is to match the processing speed to the high-speed link rates. Therefore, AP should not slow down the non-active fast path; active services for the active packets must operate at speeds comparable to the line rate (i.e., introduce minimum delay that does not impede the application performance) [SterJ].

Compared to traditional network nodes, AP capable network nodes (i.e., active nodes) require a considerable amount of processing power and memory resources. However, the recent advances in the computing hardware technologies, as predicted by Moore's Law, advocates the feasibility of introducing computationally complex services in the networks. The management of such services in the networks is determined by different levels of granularity of control. That is to say, based on the user type (i.e., service provider or customer), the AP can be controlled at different granularities through the use of active packets. In fact, the granularity of control on AP is another criterion that distinguishes different active networking approaches from each other. Therefore, the cost of introducing AP and the allowed levels of user control are two important design parameters
for any active network architecture. The following subsections describe what role these parameters play in the operation of an active network.

### 3.4.1 Cost of Active Processing

Although the cost of producing semiconductor devices is decreasing continuously, upgrading network facilities for supporting AP may come at considerable cost. Therefore, the network resources should be traded against one-another, subject to some constraints, to determine the worthiness of introducing AP [SterJ2]. Moreover, the trade-offs of introducing AP can also be considered from the user point of view, who is footing the bill for such advanced services and providing the revenue of the service providers.

The network resources that can be traded with each other are processing power, memory and bandwidth. Constraints such as latency for time sensitive applications and battery power in the context of mobile nodes should also be considered. The relationship between the resources and the constraints can be formulated as [SterJ2]:

\[
\langle P, M, B \mid L, W \rangle
\]

where \( P \) stands for processing power, \( M \) for memory, \( B \) for bandwidth, \( L \) for latency, and \( W \) for power. The formula represents the trade-off between processing, memory, and bandwidth, constrained by the latency requirements of applications and energy capacity of self-powered mobile nodes. However, if user perceptual satisfaction is a design factor in the introduction of AP in the network, then the user perceived QoS should also be included in the mapping relationship between the resources and constraints; hence the formula becomes \( \langle P, M, B \mid L, W, Q \rangle \), where \( Q \) stands for QoS. This representation shows that the constraints can be traded against the three fundamental resources. For example, a rate-controlled transcoding operation would require \( P \) and \( M \) but in turn would reduce \( B \), while enabling better \( L \) and \( Q \) performance.

The interrelation between the network resources (i.e., \( P, M, B \)) which can also be associated the cost of AP. Given the limitations on \( P \) and \( M \), the efficiency of performing AP increases as the link speed decreases. In [PlatB], this is represented by:

\[
\frac{(P+M)}{B}
\]

which means that the \( B \) is inversely proportional with the cost of increasing \( P \) and \( M \) in the network. Consequently, AP can be more beneficial if the \( (P+M)/B \) ratio is high (i.e., on lower link speeds at the network edges) than very high speed links (e.g., 10Gbits/s network core). This can be illustrated with the following example:
• Assume an active node with a commercially available microprocessor like Intel® Pentium® 4 with 3GHz clock speed, operating on a typical 10Gbits/s backbone link. Let's also assume that active packets carrying some multimedia data with average packet size of 512bytes is to be processed. This means that the router has to forward around 2441406 packets-per-second, sparing around 1229Hz for each packet. If the link speed was 155Mbits/s at an edge router with similar characteristics, then the processor could spare around 79277Hz for each packet. As a result, it can be argued that given a specific \((P+M)\), AP is more likely to make an impact if \(B\) is small.

Therefore, these factors should be taken into account when designing an active network and placing AP capabilities in its nodes.

### 3.4.2 Granularity of Control

Active networking approaches can be distinguished between each other with the granularity of control they provide on AP. The scale at which “user” packets are allowed to control AP in an active node is called the granularity of control. In this context, the term “user” refers to any authorised entity who is allowed to inject code in the network to customise the behaviour of the AP capable nodes. At one extreme, it is possible that active node behaviour is modified by a single packet executing in the node and affecting all other subsequently arriving packets at the node. At the other extreme, every packet may instigate a particular active service at the arrival and terminate that service as it leaves. However a more practical approach would require a number of packets installing some state in the node which all other packets belonging to the same flow (i.e., having a set of common characteristics) would mutually benefit.

The control of AP can be performed at four granularities; global control plane, per-flow control plane, per-packet control plane and per packet data plane [SterJ2]. This affects how and when AP must occur, ranging from a single packet to a burst of packets, to per flow or connection. An active network, depending on its design objectives, may allow for AP at any control granularity. For example, an active network architecture may allow programmability all the way from global control plane to per-packet control plane, but not necessarily at the per-packet data plane.

In terms of granularity, the coarsest AP control takes the form of the global control plane functions, which represent the processing that affects the active node operations as a whole, and perhaps the entire network. Such functions may be used for monitoring the traffic patterns and network health, which are useful for customising the network management policies and traffic engineering functions for optimised network operation. Per flow control plane functions operate
on individual flows and provide the connection maintenance and flow control based on the requirements of the network operator. The fine granular control takes place at the per-packet control plane. Such control allows the AP to distinguish amongst different packets in a flow and selectively manage them based on a user defined criteria. An example application to such functions is AP that enables selective packet filtering in a traffic flow. At the finest granularity of control, per-packet data plane functions operate on the individual packets and may modify, create or destroy them. A typical example to this kind of control is a transcoding operation. In this kind of operations, the resource requirements of AP in terms of $P$ and $M$ is higher then other AP control approaches since the execution speed should match with the line rate. Provided that the users are authorised, the control of AP can be provided at combination of these granularities.

3.5 Deployment Strategies

Introduction of active networking requires considerable modifications to the existing network hardware. This can be realised in the form of either replacement or upgrade of existing network nodes. It is important to note that there is not a single way of deploying active networks. The deployment can be overlaid or embedded depending on the need and choice of the network operator [SterJ]. On the other hand, the choice for the location of the deployment is an important design factor especially if a partial deployment strategy is followed [SivaR]. Therefore, although the design objectives for all active networking models are similar, there are differences in deployment strategies followed. In this section, the existing views about how and where to deploy active networks are discussed.

3.5.1 How to Deploy

For enabling active networking functionality in a network, it is required to choose a deployment strategy, which provides the network customers with the necessary AN interface so that they can benefit from AP in the network. One way of deploying active functionality into the network is to build an overlay active node formation on top of the existing network infrastructure. The other way is the embedding approach, where active nodes are progressively placed amongst the existing network nodes.

The overlay architecture can bring add-on functionality to networks where it is difficult to change the structure of the network and the technology of its nodes. For instance, the overlay system could lie on top of an existing IP network, whereby the active routers are simply implemented as user-level AP engines. In this scenario, active packets need to be explicitly addressed to active nodes which are interlinked by means of a tunnelling mechanism. This approach is useful for
testing of different execution environments and active services (i.e., for research purposes) without impeding the operation of the existing network infrastructure. In other words, it allows a controlled introduction of new services in the networks. For this reason, major active networking research activities have all followed this kind of deployment strategy [ABONE]. The downside of this approach is active devices need to be fully incorporated into the network and its capabilities may be limited by the underlying traditional network.

Alternatively, the embedding (partial or full) approach stands as a more flexible deployment strategy since active nodes can selectively be deployed where they are most required in the network. The gradual replacement of traditional nodes with the active ones is a more cost efficient way of enabling AN functionalities in the network. In this scenario, active nodes may use tunnelling to directly communicate with each other or sometimes may even make use of the non-active parts of the network for forwarding packets. For such cases, an abstraction of the non-active path should be made available for active nodes so that compensatory measure may be taken to account for the possible performance loss over the non-active path [SivaR].

### 3.5.2 Where to Deploy

Packet networks can be represented with the cloud model which is composed of core routers in the centre and edge routers at the borders [CarpB]. Core routers perform the transportation of large volumes of traffic and forward at much higher speeds than those at the edges. For this reason, they are traditionally required to be stateless, mainly due to their limited P and M resources. Considering the advancements in the integrated circuit technologies, it may claimed that the routers are becoming faster, hence it should be possible to allow some state information reside in the core network. However, the traffic volumes networks need to carry are growing as well, and the demand for bandwidth is ever increasing. In addition, upgrades in the core of the network are usually more expensive than those at the edge network. Therefore, it can be argued that the stateless nature of the core network is not expected to change dramatically in the near future. On the other hand, edge routers carry less volumes of traffic and are able to maintain state information for traffic flows and perform traffic conditioning operations.

In the light of these facts, the network operator has to decide where to place its active routers for optimum performance gain. The existing AN research mainly focuses on the router architectures, programming models for AP, and possible AN applications. There is little focus on where active nodes should be deployed in the network. In [WethD], it is argued that the programmability of network nodes should not be realised at the network core, and the network core should only function as the fast IP-forwarding engine. However, based on the explanations given above, it is
sensible to locate computationally intensive AP at the edges of the networks, while allowing limited programmability at the core of the networks. Several design choices for the location of active nodes are given in [SivaR].

3.6 Applications of Active Networks

Most of the problems associated with traditional packet networks are related to their rigid and inflexible protocol stack, and installed software components that are not optimised for every user and application needs. Active networks have potential to solve problems of traditional networks and support new types of applications and services as soon as the requirement arises. However, as it is with every type of new networking approach, the widespread adoption and deployment of active networks depends on the development and demonstration of valuable applications and services that demonstrate its benefits to service providers and the customers (i.e., end-users). Although the active networks research is ongoing and there exists no single type of active networking approach, the common goal is to identify the possible advantages of enabling AP inside the network. The research community have addressed a number of applications which are helpful in demonstrating the usefulness of active networks. These can mainly be categorised into the following areas:

- QoS.
- Protocol Deployment and Signalling.
- Network Management.

In the following subsections, specific examples about how active networking concepts can be utilised are given. The QoS related applications of active networks have been a motivational factor for the research work presented in this thesis.

3.6.1 QoS

IP-QoS architectures usually operate with predetermined network parameters to offer an efficient transport service for the users. Active networks, on the other hand, allow for each user to initiate services based on its requirements. Moreover, service providers can adjust their service parameters when and where needed to provide support for the user traffic. Therefore, active networking is a promising performance booster for the networks and multimedia applications. The QoS improvements can be introduced in the following areas:

- Routing.
- Traffic conditioning.
• Congestion Control.
• Delay and Jitter Control.
• Error Control.
• Caching and Retransmission.
• Personalisation and Context Awareness.

3.6.1.1 Routing
In traditional networks, the routing algorithms are standardised, static and deployed by the network operator. These algorithms (e.g., OSFP [MoyJ], IS-IS [CallR]), make use of constraint functions, which determine the outgoing link for a given IP packet. However, these algorithms are simple mechanisms designed to provide adequate performance with minimum complexity. Therefore, adaptation to changing network conditions is normally tackled by end-user provided mechanisms. Enabling active networking functionalities in the network routers would allow users to implement specialised criteria and different rules for the routing of their applications, and providing opportunity for service providers to charge based on the routing performance they offer [MaxeN]. Depending on the code distribution mechanism used, state information could be installed in the network for flow recognition, which would allow all of the packets belonging to the same flow to be treated with the same active routing service. The common benefits of active routing approach can be exemplified with multicast routing. In a multicasting session, short lived congestion periods may degrade the application performance and typically affect all the receivers downstream in the multicast tree from the point of congestion. Since the network topology is hidden from the users, it is not possible to determine where the problem occurs, thus all the users belonging to the same multicast session get affected by the reduced performance. In addition, the change in the network conditions or user profiles may modify the dynamics of the multicast tree, which may demand an effective and dynamic configuration mechanism for the tree structure and the packet routing protocol [LehmL]. A number of active multicast projects have realised the value of placing the functionality inside the network when it is needed at the multicast branch points [SandM], [AkamH]. Coupled with an appropriate pricing scheme, application aware active routing may help improving the performance of the applications (e.g., forwarding the less time-critical information over the longer but less congested link) [MaxeN].

3.6.1.2 Traffic Conditioning
Traffic conditioning help the network routers implement the service agreements between the service provider and the customers. It is comprised of mechanisms such as classification, scheduling and queue management. Classification helps the router to select an appropriate output queue for its incoming packet. Scheduling is used to systematically pull packets from the output queues and the queue management works on controlling the queue size based on a given set of
parameters. In traditional networks these operations are fixed, and cannot adapt to changing network conditions. If routers are made programmable, these mechanisms can be modified dynamically where traffic conditioning operations are influenced by the user/service provider requirements and/or the changing network conditions. For instance, an active buffering mechanism would consider the delay and loss requirements of an application packet and place it in an underutilised queue that can guarantee the QoS required by that packet. Similarly, an active scheduling mechanisms means that the scheduling algorithm can be modified to better serve the application requirements [HussA]. Active networking enable customisable queue management schemes where the type or parameters of the dropper algorithm can be altered to bring finer service differentiation between the traffic flows sharing the same queue [HsiaH].

3.6.1.3 Congestion Control

Multimedia streams are usually carried over unreliable transport mechanisms, which do not respond to network congestion. Therefore, they are subject to uncontrolled losses during the congestion periods. In this respect, it is necessary to reduce the response time to congestion or even predict it before it happens. This is only possible with network based mechanisms which constantly monitor the traffic load of the system and arrival patterns of the network traffic. Active networking can help applications instruct the network about their requirements and how should such traffic be handled in the case of congestion. In other words, the network is able to determine when, and applications provide the information about how to adapt to adapt the flow to the available resources. Intelligent packet discarding is a popular approach for congestion control [BhatS2], [RamaR], [SivaR], [BalaR]. For video communications, especially for multicast sessions, adaptive frame/layer discarding techniques have also been demonstrated to improve QoS [KellR], [KangS], [HeD]. Transcoding and dynamic alteration of algorithmic dropper properties are a few of the possible mechanisms that could be utilised to provide graceful quality degradation for multimedia streams [HsiaH], [SivaR], [AmirE]. Unlike the case in traditional networks, active network based mechanisms migrate among networking devices and are activated when and where needed.

3.6.1.4 Delay and Jitter Control

Active networking research has also made progress in developing strategies for controlling the delay and jitter in the networks. Traditionally, these issues are tackled with using various queuing and scheduling schemes [Juni1], and resource reservation. The resource reservation requires an explicit allocation of network resources on a particular path between the end-point (guaranteeing bandwidth and delay and loss requirements). Such allocation may result in inefficient use of network resources since the reserved resources can be underutilised by the reserving flows. Furthermore, sometimes it may not be possible to guarantee any reservation if the network is
congested. Active networking techniques can help to dynamically change the route of the packets which need allocated resources. Active network based bandwidth reservation [WillD] can find a route or routes on which the needed reservation can be made, and the scheduling weights can be reconfigured to support the necessary reservation requirements.

The task of jitter control is to make sure synchronous traffic gets through the network smoothly. Active networks can enable jitter control performed on a per packet basis, not per flow as in today’s networks. One way of controlling the jitter is to compute the expected transit time for each hop along the path. This information is carried in packets. If at a hop the packet is behind the schedule, the router can increase the priority and send it faster; if a packet is ahead of schedule, decrease the priority and send it later. Employment of complex and application dependent packet handling mechanisms for jitter control is also possible through the use of the active nodes’ programmable interfaces. In such approach, individual packets can vary their jitter control algorithm to instruct the network about the relative jitter sensitivity of packets of the same multimedia stream. For example, a packet providing visual background that is not in motion may have a much higher tolerance for jitter than a packet containing the data of a more dynamic portion of the visual or audio transmission [BushS].

3.6.1.5 Error Control

Error control is an important research subject in multimedia communications. Packet loss and channel noise can introduce errors in the decoded multimedia stream and degrade the QoS received by the users. Although, end-point based schemes can be useful in alleviating such adversities, they introduce additional complexity, require extra processing power, make inefficient use of bandwidth, and lack in scalability. Therefore, network operators may be required to utilise proxies and media gateways for avoiding such disadvantages [DoganS3], [EminS], [WaraT]. Nevertheless, these devices are statically configured and positioned; hence, cannot react dynamically to changing network conditions, user requirements and applications. Active networking research, although highlighting the necessity of such complex computations in the network, have not produced extensive amount of research in this field. The existing research mainly focuses on adaptive FEC schemes that can be introduced at the active nodes depending on the network conditions [LechC], [MeggJ]. It is in the belief of the thesis that active networking services could include various other network based mechanisms that can be useful in improving the QoS of the multimedia applications.

3.6.1.6 Caching and Retransmission

Caching and retransmission are interrelated techniques that are effective in improving the perceived QoS. The caching of most frequently accessed objects in locations near to end-users is a
useful technique for saving the network bandwidth and providing fast response times for applications in use. Users are either forwarded the requested object without needing to communicate with the object source or retransmitted any lost information from the caches. Caching schemes require decisions about where to locate objects and how to forward requests between caches. In today’s networks, caches are statically configured (limiting the ability to react to dynamic conditions) and incur administrative burden. Active networking can add flexibility to the caching mechanism by routing the cache requests to pre-configured cache locations, and combining small caches with information about the contents of nearby caches, at each network node [LegeU], [BhatS3].

In traditional networks, multimedia streams are transported over the unreliable transport protocol (i.e., UDP) that does not employ any retransmission mechanism for the recovery of lost data. Instead, multimedia applications are encoded with some redundancy, which allows for acceptable perceptual quality even in the case of some packet loss. The reason why retransmission is not employed in real-time and low-latency video communication applications is due to the excessive end-to-end delay. Active networking can minimise the retransmission delay by allowing active nodes to be programmed to perform retransmissions when and where it is feasible (e.g., from the nearby caches in the case of link congestion). Two AN-based retransmission schemes performed in a layered multimedia multicast [LehmL] and on a point-to-point basis [BrasJ] have demonstrated the possible advantages of having intelligent caching mechanisms in the network.

3.6.1.7 Personalisation and Context Awareness

Personalisation of the multimedia services involves the adaptation of the content to the limitations of the user terminals and the transmission channel conditions. The challenge for such adaptation functionality is hence primarily dictated by the bandwidth requirements and device capabilities of the subscribers to some networks [DogaS]. Furthermore, personalisation also refers to the manipulation of multimedia information based on the user context, which defines the personal preferences and the environmental variables of the user’s surrounding. Unfortunately, there is little effort within the active networking research community to develop architectures that enable personalisation of multimedia services through dynamic interactions between the user and the network. The work presented in [OttM] gives an example about how active nodes can be used to adapt a multimedia stream to user terminal capabilities without hindering the continuous display of the content. In addition, the transcoding services presented in [EminS] can be considered to be a starting point for developing low complexity/latency network services for personalised multimedia communications in the context of active networks.
3.6.2 Protocol Deployment and Signalling

The initial thought in active networking support for introduction of new functionality in the network through dynamic protocol deployment. This approach opens up the network and allows third-parties to install, test and release new services and protocols within a very short time without the need for standardisation. The active bridging project [AlexD2] is one of the first active network projects, which demonstrated how it is possible to upgrade/replace network protocols on-the-fly. In [YamaL], an adaptive multicast support protocol was proposed which enables selective filetering of data packets, and pruning of multicast layers to account for different user terminal characteristics and varying network conditions. Furthermore, [LegeU] showed the possible performance improvements of distributed applications by using active protocol deployment.

On the other hand, active network based signalling research mainly focuses on deployment of complex control-plane functions that can provide flexibility in network management and handling of deployed protocols. In related research, [BradB] explores the feasibility of utilising a special EE for signalling, and demonstrates the potential of active networks for dynamic deployment of protocols and management functions in the network.

3.6.3 Network Management

The traditional network management architectures are known to be inefficient due to their centralised nature. This means that management functions are highly complex (depending on the network size), waste network resources (due to high volumes of control traffic) and cannot provide a dynamic network state when and where needed. Therefore, they suffer from scalability problems. Distributed agent-based management schemes, such as CORBA, have been designed to alleviate these problems [CORBA]. Nevertheless, these too make inefficient use of network bandwidth and lack in the ability for agents to communicate with their neighbours for effectively carrying out distributed management tasks. To overcome these disadvantages, an active network based distributed management framework was proposed, which allows the distribution and execution of network management applications in the active routers [RazD]. According to this research work, the use of active networking techniques allow fast and easy deployment of distributed network management applications in IP networks.

The active networking based network management enables the creation of self-configuring, self-diagnosing, and self-healing networks. By this way, it is possible to capture the state of a node by sending a single smart packet that gathers all relevant information at one time. For instance, network operator's smart packets might travel inside the network executing control-plane routines inside the active nodes. Hence, the active nodes can send state-update messages to the network.
operator and the neighbouring nodes, enabling them to take necessary actions for management of
the network when and where needed, without producing large volumes of control traffic [CalvK].

3.7 Technical and Practical Challenges

So far in this chapter, the active networking concept as the evolution of traditional packet
switched networks has been briefly introduced. However, active networks face with a number of
technical and practical challenges that need to be tackled in order to make a significant impact in
the way networks operate today. Technical challenges are made up of technological problems
which hinder the feasibility of deploying active networking components into the packet networks.
Practical challenges are related with the way network operators and hardware vendors like to see
and manage the networks, and whether they could be convinced to invest on this technology.

3.7.1 Technical Challenges

Although there is on-going research going in the field of active networks, there are certain
technical challenges with active networking technology that needs to be addressed. The most
notable challenges involve interoperability and practicality.

Interoperability refers to the level of compatibility between different active networking
approaches, and between the non-active parts of the networks. The research activities in active
networking field, although following similar principles, are somewhat divergent in their
approaches to active packet, active node, and the programming model designs. Different institutes
are working on different architectures, which are usually tailored for specific applications; hence
fail to establish a generic model that can provide comprehensive programmability for all sorts of
applications and network topologies. For successful widespread adoption of this technology, it is
vital in future that different active network approaches can interoperate through common network
programming interface. Consequently, it may be required to have the active node architecture as
well as the active packet structure standardized. Even if a generic active network model is
produced, the next problem would be how to enable interoperability between the active and non-
active networks. This can be explained with two examples: Firstly, if the non-active portions of
the network are within the administrative domain of the same network operator, then the active
nodes would require an abstraction of the state of those portions of the network for the services
that reside in the active nodes [SivaR]. Secondly, if active packets need to traverse another
operator's domain, then there is a need to map certain network level parameters between domains
to maintain the service guarantees given to the users of the active domain. This is analogous to
bilateral agreements existing between the operators of different domains in today's networks.
Practicality can be related to safety, efficiency, and flexibility. If a system can demonstrate satisfactory performance in all three aspects, then it can be regarded as practical [MoorJ]. Since active packets are used to deploy new protocols and services into the network, a safety and security mechanism must be employed to acquire protection against malicious code (prevent misuse of network resources). However, as the safety mechanisms are enhanced, the flexibility of system may be reduced since users would be restricted in their network programming abilities. On the other hand, the system must operate at link speeds (do not introduce unacceptable delay overhead), while being able to offer rich AP functionality. Today’s IP networks can be considered as the lowest threshold to how much safe, efficient and flexible active networks should be. Naturally, for active networks to be considered practical and deployable, they must provide distinctive advantages in terms of safety, efficiency and flexibility in comparison to traditional IP networks.

3.7.2 Practical Challenges

Developing a technology which can demonstrate its effectiveness on experimental test-beds does not always mean that it will be realised in real-life. Although active networking research has not reached to its maturity, several practical challenges ahead of its adoption in real life can easily be foreseen. The major challenges are associated with network hardware vendors, conservative service providers, and finding the killer application [SterJ2].

Because of its programming requirements for dynamic provision of protocols and services, active networking needs to utilise open network architectures and signalling. It is anticipated that network hardware vendors do not have a positive approach to the idea of opening their hardware interfaces or expose their signalling and routing code [SterJ2]. Only a strong demand from the users and service providers may motivate the vendors to provide the necessary open interfaces and signalling. Nevertheless, in [BaiY] it is claimed that the industry has realised that AN technology may become a significant revenue source. As a result many companies, such as Nortel, CISCO, and Lucent, have been very active in the research and development of active routers. This can be seen in the current trend of implementing programmable network processors in routers to support the increasingly sophisticated packet classification and scheduling desired by network service providers or network operators.

While service provider motivation is needed for the motivation of vendors to support AN deployment, some service providers can be extremely conservative with a short planning horizon that tries to meet predicted user demand by investing on the network capacity. They are not able
to comprehend the new possibilities that AN would provide in terms of network management, and service deployment, which can save them time and bandwidth, as well as providing new sources of revenue (i.e., selling active services). The main discouraging factor for such service providers is the potential cost of deploying active networking, which could prove to be considerably high.

The final major practical challenge is finding the killer application. Killer application is a jargon used to describe the application of significant drive in the development of a networking architecture. For instance, in the Internet this has proved to be web services and SMS/Voice in wireless communications (GSM). Therefore, AN needs also must find a killer application that will drive its development and widespread deployment. However, the emergence of a killer application is difficult without sufficient network infrastructure [SterJ]. Active networking research community has been trying to highlight the dynamic protocol and service deployment as the potential killer application for AN by demonstrating its usefulness in various applications. Yet, it is difficult to convince conservative service providers that AN is the next step forward in networking and they ought to adopt it sooner or later.

3.8 Concluding Remarks and Recommendations

This chapter has made a brief introduction to the Active Networks. Active Networks are seen as the natural evolution of traditional packet networks where both network nodes and end-users are equipped with intelligent tools. This approach, although inherently complex and requiring significant computational and memory resources in the network nodes, can bring the flexibility and development speed required by the network operators, and especially by the network subscribers. The fundamental entities of active networks are the active packets and the active nodes. Users can dynamically program and modify the operation of network nodes depending on their application needs. Within the concept of active networking, network administrators gain a great deal of flexibility where they can charge according to the services required by the users. In addition, they can dynamically create, access, and modify the state of the network.

As AN requires AP functionalities to be installed in the network, it is claimed to be contradicting with the end-to-end arguments. However, AP allows users to implement precisely the service they need and improve the end-to-end performance of the network; a property in agreement with the end-to-end arguments. Therefore, applying end-to-end arguments should be carefully reviewed and not applied in a definitive way. Instead, the specifics of each particular active networking idea would benefit from evaluation in the light of the end-to-end principles. It is in the view of the research presented in this thesis that it is scientifically reasonable to question the applicability of
end-to-end arguments and seek for alternative approaches for future networking systems. Therefore, no networking approach should be absolutely confined within the limitations of any kind of arguments, theories or rules, yet end-to-end arguments are useful starting point to the design of a new system like the Internet.

Currently, there exists no single AN model with a generic interface that can be accessed and programmed by various types of user programming models. The research in AN is still at its experimental state and the commercial adoption is far from realisation. A number of AN frameworks proposed in the literature in various ways; active packet format, code distribution mechanisms, code encoding type, NodeOS and EE characteristics, deployment strategies, interoperability, network layer on which they place AP (i.e., application, transport, network, medium-access or physical), and the applications they target. Therefore, in this thesis, no particular active networking approach has been elected as the base model. Instead, the use of active networks is based on certain assumptions that are elaborated in Chapter 5.

Important challenges ahead of AN adoption have been listed in Section 3.7. In the view of the research presented in this thesis, QoS related applications are the ones that are most likely to motivate service providers and vendors to adopt active networking technologies. Effective QoS provision techniques are central to them since they imply new and better ways of deploying services in the networks and charging their customers without the need of long and cumbersome standardisation process. In this scenario, service providers will be able to sell services rather then bandwidth, and at the same time have dynamic control of their networks. On the other hand, users will be able to choose the services they like, and pay according to the exact QoS they receive. Another advantage for the users is that third party service developers will be introduced in the market, which will create competition amongst different service providers. Therefore, users will be offered better services for lower prices.

It is interesting to relate the active networking technology to future networks and user profiles. The author of this thesis believes that the future networks will be formed of various access networks linked to a group of service provider networks, which are all interconnected through a network core made of very high speed links. Advances in the semiconductors technology will enable extensive processing capabilities to reside in the networks. Therefore, the network edges will be fully processing capable for enabling complex computations to take place. The network then will be seen as the *service delivery medium* rather then a data transport medium. Users will be fully immersed into multimedia applications through the use of various access devices with different capabilities. Further, the network will have a detailed knowledge of the user's context and provide an interaction platform for users to input their preferences. This networking scenario
Chapter 3. Active Networks

will require the network nodes to be fully programmable and intelligent robots rather than a policy managed, standard compliant static forwarding machines connected via high capacity links. AN is the major step forward in the evolution of networks towards this direction.

Recognising this fact, this research tries to integrate active networking QoS capabilities with a promising IP QoS architecture (i.e., Differentiated Services). Migration to fully active QoS schemes can be realised by first bringing additional support to existing approaches through the use of the AN approach. Eventually, after this transition period, it is expected by the AN research community that AN based schemes will dominate in the networks. Therefore in this thesis, the possible advantages of utilising AN services over Differentiated Services architecture are also studied.

AN is about creating a new networking notion where the end-points area able to establish dynamic interaction with their networks, and program the network behaviour according to users'/applications' requirements. In this respect, it encompasses a wide scope of research fields that are related to telecommunications, and cannot be all covered within a single research project. The intention of the research work presented in this thesis was to explore the possibilities of improving multimedia QoS with the use of AP inside the networks. Therefore the focus is put on the multimedia communications related issues rather then the technologies enabling AN. The next chapter develops the background information about Differentiated Services, and demonstrates its benefits in terms of providing QoS for multimedia communications. In the later Chapter, the active services which were developed in the context of this research work are presented. The advantages of deploying such services are examined for both fixed and wireless networks.
Chapter 4

4 Differentiated Services

4.1 Overview

Low-latency and real-time multimedia applications demand more stringent QoS requirements than traditional applications which are transported using reliable protocols (i.e., TCP). Since these applications are usually carried over unreliable transport mechanisms (i.e., UDP), their tolerance to packet loss is extremely limited. However, the Internet is an unpredictable and time-varying channel. Therefore, unless the network resources are sufficiently over-provisioned, the best-effort Internet model is not sufficient in meeting the demands of such multimedia applications.

Differentiated Services is the IP QoS architecture which aggregates traffic streams of similar characteristics into a number of distinct traffic flows. These flows are then provided with differentiated treatment and forwarding as they traverse the network. By this way, in the case of network congestion, certain flows would be offered prioritised treatment over the others. In addition, the DiffServ architecture is also able to provide specific delay/loss bounds to the required traffic flows. Consequently, DiffServ has been a popular subject of research for providing QoS enabled transportation of multimedia streams.

The aim of this chapter is to introduce the DiffServ concept and demonstrate its possible advantages in multimedia communications. It starts with giving a brief overview of the architecture and introduces the concepts behind its motivation. This is followed by the definition of the architectural components of a typical DiffServ network. These traffic control mechanisms are the main elements of the DiffServ architecture which enable differentiated QoS provisioning. A typical example of how different applications can be group into distinct DiffServ aggregates is given in Section 4.5. The DiffServ architecture has also been considered in the design of the wireless networks. Therefore Section 4.6 introduces the relevance of the DiffServ concept to UMTS. The rest of the chapter is dedicated to the application of DiffServ concept on the multimedia traffic. In Section 4.8, the advantages of introducing service differentiation in the networks are demonstrated through a series of experiments. The chapter finalises with the concluding remarks and the recommendations based on the investigations carried out in this research.
4.2 Conceptual Overview

4.2.1 Motivation

The most promising approach for providing service guarantees in IP networks is the Differentiated Services initiative. The Differentiated Services Architecture [BlakS] has been designed to provide network services with QoS guarantees in the Internet by introducing relative priority treatment scheme to different classes of traffic. There are two major drivers for the development of such architecture. The first one is the stringent requirements of the real-time multimedia applications. The other major driver for network QoS is the need for service differentiation due to competitive nature of the marketplace. An effective QoS architecture must provide a means for specifying performance objectives for different types of packets/traffic as well as delivering those performance objectives. Even in enterprise networks where sufficient bandwidth is available, it may be necessary to differentiate the real-time traffic. That is to say, the voice traffic requires a service level with relatively low bandwidth but more controlled delay variations and is somewhat robust to network congestion. Therefore, service differentiation is essential for accommodating various applications with heterogeneous requirements, different user expectations, and for facilitating a service based pricing mechanism for the service provider. In other words, if practical methods for differentiating traffic in IP networks exist, service providers may offer services with defined performance characteristics that are cost linked to certain quality of service definitions. Further, with such IP-QoS architecture, more and more communications can migrate to the Internet, increasing the service providers’ revenue and decreasing the network costs [CarpB].

4.2.2 Service Model

The DiffServ architecture does not define which traffic treatment mechanisms are required but instead it makes fundamental assumptions about the structure of the network and defines how QoS building blocks can be put together to create an architecture which can deliver the desired services to the customers. The architecture was developed based on the recognition of the fact that specific service guarantees can only be offered within the administrative domain of a single network operator. This refers to an edge-to-edge service model across a single domain where the QoS to be received is defined with an appropriate service agreement between the users and the service provider. However, end-to-end service provision may require the user traffic to traverse multiple domains of heterogeneous administrative control and technological structure. In that case, an inter-domain QoS mapping mechanism should be in place to ensure the construction of end-to-end services.
Within the administrative domain of a service provider, the services that are to be provided to a customer are specified by a contract that exists between these two parties. In DiffServ architecture, this contract is named as Service Level Agreement (SLA). SLA specifies the forwarding service a customer should receive [NichK]. It is a technical term encompassing technical features and parameters of the service as well as legal and charging aspects [GodzJ]. A part of these technical features and parameters of the service are the traffic conditioning rules (traffic conditioning), which are stated as Traffic Conditioning Agreement (TCA). This is an agreement specifying classification rules (traffic classification) and traffic profiles in terms of their temporal characteristics, such as rate and burst size. The customer is required produce traffic profiles conforming to the definitions provided in the TCA. This is enforced by defining certain traffic shaping and packet discard rules, as well as employing admission control mechanisms at the network entrance. Overall, TCA defines all of the traffic conditioning rules explicitly specified within a SLA along with all of the rules that are implicit from the relevant service requirements and the DiffServ domain’s service provisioning policy [NichK].

Recently, Internet Engineering Task Force (IETF) decided that the definitions of SLA and TCA were too broad and covered considerations that were of business nature, as well as those that were highly technical. Therefore, in the interest of separating the technical aspects of the contracts from their business features, new terminology was introduced. In this regard, Service Level Specification (SLS) and Traffic Conditioning Specification (TCS) terms were defined [GrosD]. SLS is defined as a set of parameters and their values, which together describe the service offered to a traffic stream in a DiffServ domain. TCS is a subset of an SLS and specifies the technical service performance parameters for each QoS level and the traffic profile associated that level. These parameters may include drop probability, throughput, latency and other classifier and conditioner parameters.

An SLA (hence SLS) can be static or dynamic. Static SLA means that the network services for the users are predefined and the network nodes providing these services are pre-configured. In static configuration, traffic aggregates are assigned static membership and their limitations on accessing network resources are defined. During the call setup, a certain portion of these resources is assigned to an individual flow throughout the call duration. When the provisioning is static, usually with time-of-day specifications, it cannot be changed with response to a dynamic message. On the other hand, with dynamic SLAs, the network services can be negotiated and resources be allocated on demand and per-call basis. However, this requires a signalling protocol (e.g RSVP) to be used [XipeX]. Nevertheless, signalling is not built into the DiffServ architecture and is not a part of its standardisation activities, thus whether to use static or dynamic SLA is left to the network operator’s decision.
Based on the specifications of the agreed SLAs, the DiffServ compliant routers in the DS-domain are configured by the network management mechanism. This sets up the QoS behaviour table inside the router by translating SLS and TCS specifications into router configuration parameters. In the same way, domain boundaries are programmed to form the traffic aggregates (TA) from traffic streams, which have similar QoS requirements. The QoS specific forwarding treatment applied at a DiffServ compliant network node is called Per-Hop-Behaviour (PHB). In a DiffServ router, different traffic aggregates are treated with different PHBs. Within the DiffServ domain, all the nodes implement the same PHB mechanisms and operate with a common service provisioning policy.

It is important to define the transit expectations of traffic aggregates across a network cloud (DiffServ domain). This helps in the deployment of QoS services inside the network and assists in cross-cloud services and agreements to be composed. Therefore, IETF has adopted the phrase "Per-Domain-Behaviour" (PDB) to describe QoS attributes across a DiffServ domain. In other words, PDB defines the expected edge-to-edge treatment a traffic aggregate is expected to receive [NichK2]. PDB parameters are specified in SLS. A single PDB is defined for each QoS level that the DiffServ domain can offer. Therefore, PDB represents the overall effect of traffic classification, conditioning and PHBs on the traffic aggregates. PDBs have measurable and quantifiable attributes that can be used to determine what is most likely to be experienced by packets as they traverse the DiffServ domain. These attributes are generally expressed as bounds or percentiles. For example, a loss attribute may be expressed as "for QoS level A, no more than 0.2% of the packets will be dropped".

Although the DiffServ service model defines the agreements, specifications and traffic handling tools, it makes no clear definition of how network resources should be shared amongst competing traffic aggregates. SLAs may state which QoS level (or behaviour aggregate) gets what percent of the overall resources. Nonetheless, there is no specific reference to how does the traffic aggregates of the same QoS level share the available network resources. In addition, it is also important to be able to use the underutilised resources that are otherwise reserved for different QoS levels. An effective way of handling this problem is to use admission control [MykoN][ZhanZ][KnigE]. Admission control can help to uphold the organisation policies, hence assure high quality communication by ensuring the availability of resources to handle a traffic load. In particular, real-time multimedia flows, which are usually unresponsive to the fluctuations in the network bandwidth, can benefit from the use of admission control mechanism. Through the use of signalling protocols such as SIP, H.323 and RSVP etc., applications can negotiate admittance and make efficient use of network resources. Having assured a negotiated input rate, the network may
make use of prioritised traffic handling mechanisms to ensure nominal delay, jitter and loss for the transmission [BakeF]. By this way, the already admitted traffic streams do not anticipate any deterioration in their agreed QoS level.

The admission control can be provided by agents that have some knowledge of organisation’s priorities and policies such that they can allocate the bandwidth and mark the packets accordingly. One approach of doing this is to utilise Bandwidth Brokers (BB) (or allocators) that can be programmed with the organisational policies, keep track of the current allocations of the marked traffic flows, and interpret new requests to remark the traffic flows in the light of new policies and current allocation [NichK3]. Especially in the case of dynamic SLA usage, the BB component becomes the essential part of the architecture since the resource allocation and QoS level assignment is performed on demand.

4.2.3 Scalability

Unless the QoS mechanisms employed are simple and easy to deploy, the successful implementation of the architecture would be subject to long and cumbersome standardisation process. Scalability and simplicity have been the main emphasis on developing the DiffServ architecture. Scalability is required to meet the demands of ever-growing number of applications and end-users. To enable this, the DiffServ approach assumes a stateless network core where routers handle multi-gigabit/terabit traffic loads based on a few predefined PHBs. DiffServ places the complexity at the boundaries of the network and allows sophisticated combinations numerous simple services to be available for various demands of the end-users. It is not required to deploy the DiffServ capability throughout the network operator’s bandwidth. It is possible to make a progressive deployment starting at the network bottlenecks. Several network nodes can be customised to differentiate the user traffic and provide graceful degradation to less significant traffic flows, hence allocating more resources to time-sensitive or higher-paying customer traffic.

4.2.4 Packet Differentiation

QoS in a DiffServ network provides a means for giving some packets better treatment than others based on the policies and the preferences of the organisation that is footing the bill for the required network services. However, this treatment does not tie particular applications to particular levels of QoS, as this would limit the flexibility of the architecture. Network resources are allocated according to policy, on long-term basis, and availability, which may be on short-term basis. There is an access control mechanism in place to grant or deny the QoS requests made by the users.
Chapter 4. Differentiated Services

To be able to give different treatment to different flows, the network must be able to classify packets (classification). The classification process enables the mapping of packets into specific traffic aggregates. Each traffic aggregate is then policed, shaped, queued, and scheduled according to a specific policy. Scheduling also provides means for measuring, monitoring and conditioning packet streams to meet requirements of different QoS levels. These mechanisms are implemented in the packet-forwarding path.

Applying different QoS treatment to different packets requires the network to keep a level of state information inside the network nodes and this increases the complexity. The number of ways in which packets can be treated in the forwarding path is limited. Therefore, packets flows should be aggregated according to their common packet treatment requirements (or QoS requirements) in order to reduce state and complexity in the network. As a result, the problem of providing QoS treatment to individual packet flows is simplified to treating aggregated flows. Nevertheless, the implementation of QoS mechanisms on the packet-forwarding path should not hinder the optimum operation speed of the network links.

A flow's DiffServ requirements are designated by its packets' IP header field markers. The 6-b field in the IP header is known as the Differentiated Services field (DS field) [NichK1]. It is marked with a specific codepoint called DS-codepoint (or DSCP). The network routers recognise this field in each packet and decide on the appropriate treatment. The 6-b DS field can contain up to 64 different binary values. Although it is unlikely to have 64 different packet-forwarding treatments within a network, extra space is implemented for experimentation and room for innovation as well as local optimisations. The packets, depending on the network policy, may be marked either at the original source of the traffic or at the boundaries of the DiffServ domain.

When a packet enters a router, the routing logic selects the output port and DSCP value is used to move the packet to a specific queue or to treat the packet at that port [NichK1]. The particular handling depends on DSCP’s corresponding packet-forwarding treatment. The way at which packets are treated is configured by the network management mechanism, which sets up the QoS behaviour table inside the routers. This table helps the router to place particular traffic aggregates into specific queues whose parameters are preconfigured by the network management mechanism.

4.2.5 Providing End-to-End Services

Networks are composed of administratively and technologically diverse clouds. Access to QoS can only be guaranteed at the local level if there are no service level agreements and a means of QoS mapping between the networks. Aggregate flows are composed based on local rules and the
traffic characteristics are checked at the network boundaries. When all the QoS requests are classified into a small number of aggregates, packet traffic crossing the boundaries need only be classified, monitored and measured on this smaller scale, thus simplifying the state information that is required for each flow. Although DiffServ does not attempt to give explicit end-to-end guarantees, it specifies the way in which end-to-end QoS can be constructed. For this purpose, each adjacent network has to employ a mechanism for policy exchange and establish a QoS translation mechanism with each other. This is a scalable and amendable approach to local policy control and logical way of providing end-to-end services between end-to-edge networks (through the use of bilateral agreements). The agreements made between the network operators define the policing of the traffic flows at the network borders. Through the use of bilateral agreements between the network boundaries, only a small amount of state information pertaining to the flow of packets between the clouds is required. These agreements are translated into router configuration parameters by the QoS policy management system in each network domain. It is in the interest of the sending cloud (upstream) to conform to the agreements made with the receiving cloud. Consequence of violating the rules, i.e. sending higher rate than specified, would be the reshaping of the incoming traffic that may result in a setback in the promised QoS performance.

4.3 Differentiated Services Architectural Model

The differentiated services architecture is based on a model where a network domain with defined boundaries classifies the traffic into various categories and applies conditioning and forwarding functions accordingly. This domain is named as differentiated services domain (DS-domain). A DS-domain is composed of boundary and interior nodes. The boundary nodes are responsible for classification and traffic conditioning. These boundary nodes are either ingress or egress nodes. Ingress nodes reside between the access network and the DS-domain. The traffic enters a DS domain at an ingress node and leaves the domain at an egress node. The main complexity of the DiffServ architecture resides within these nodes. Ingress nodes ensure that packets flows that traverse the DS-domain are appropriately processed to conform to the service level agreements existing between the network operator and the users. Egress nodes interconnect the DS-domain to other DS or non-DS-capable domains by reconditioning the traffic (if need be) based on bilateral agreements between two domains. If several DS-domains are interconnected in this way, a differentiated services region is formed (DS-region).

Based on the SLAs between the users and the network operator, a particular DSCP value is assigned to each user’s outgoing traffic. At the entrance to the DiffServ domain, individual traffic streams, which require similar forwarding behaviours are aggregated together to compose traffic
flows. The interior nodes (routers) are responsible for identifying these traffic aggregates through their DSCP and forwarding them according to the per-hop behaviour associated with each DSCP value. They do not keep any state information regarding to the flows but are preconfigured to associate certain DSCP values with certain forwarding treatments. This is performed by making use of a look-up table, which contains predefined behaviour instructions for every QoS level. Moreover, multi-field classification is also possible using other IP header information [BlakS]. Figure 4.1 depicts the DiffServ architecture. The customers access the differentiated services domain through an access network, which represents the local network domain that provides the connectivity. Access networks may prefer to employ admission control mechanisms to ensure the efficient use of network resources and provide better QoS.

This section introduces the functionalities associated with the DiffServ domain nodes. It starts with giving a brief overview of the border nodes. Following that, the core router characteristics and typical forwarding behaviours are presented.

Figure 4.1: The DiffServ Architecture

4.3.1 Traffic Classification and Conditioning

Traffic conditioning and classification mainly takes place at the border routers [BlakS], [BernY], [BakeF2]. It is the general term used for the operations of functional blocks of these routers. These functional blocks are:

- Traffic classification elements
- Metering functions
- Marking, Absolute Dropping, Counting and Multiplexing actions
• Queueing elements with Algorithmic Dropping and Scheduling

Amongst these functional blocks, queueing and scheduling are the two elements that are common to all routers in a DiffServ domain. Computationally complex operations such as classification, metering, marking, shaping and dropping are mainly performed at the network edges (border routers). Figure 4.2 shows the logical structure of a packet classifier and traffic conditioner inside the border routers. As can be seen in this figure, the first operation that takes place inside the router is the classification process where incoming packet stream is analysed, and accordingly classified into \( n \) number of different streams. The meter block performs the monitoring of the traffic streams and sends control information to the other blocks. Classified packets are passed to the marker block, which marks the DS-field on the packet headers based on the agreed policies. Packets are then forwarded to the shaper/dropper block for conformance to some defined traffic profile. Shaped traffic flows are queued and scheduled according to their QoS requirements. This is followed by sending the packet flows (or behaviour aggregates) through the DiffServ core, where they are subjected to forwarding treatment defined by PHBs.

4.3.1.1 Classifier

The traffic classification process is the application of the traffic identification policy at the ingress of a DiffServ domain. This process, configured by the service provider, determines which packets should be mapped onto which behaviour aggregate, and how they should be conditioned. Packet classifiers are composed of packet filters that make selection based on either the DSCP (behaviour aggregate classifier) or the value of one or more header fields (multi-field classifier). In the core routers of the DS-domain, classifiers are responsible for making simple behaviour aggregate classification decisions. However, at the edges of the domain, traffic classifiers are more complex.
Classification process at the network edges involves identification of different traffic streams, checking their DSCP and other IP header contents, exchanging information with the metering component, and if required reclassification of the traffic flows. After the classification process, packets are fed into the traffic conditioner.

### 4.3.1.2 Meter

The meter block measures the temporal characteristics of the classified traffic flows against the traffic profile specifications indicated in a corresponding TCS. The state information obtained from the meter is supplied to the conditioning blocks to trigger traffic conformance operations. The meter feeds information to both the marker and the shaper/dropper blocks. If the meter decides that a particular traffic flow is out-of-profile, it may change the marking policy and/or shaping/dropping parameters, and consecutively the queueing processes. Typical meter types that can be used are Average Rate Meter, Exponential Weighted Moving Average Meter, Token Buckets Meter and Null Meter. The details of these meters are given in [BernY].

### 4.3.1.3 Marker

Having had the incoming traffic streams classified and metered, the marker sets the DS-field of the IP packet to a specific codepoint, which specifies the particular treatment that the flow will receive inside the DiffServ domain. If the packets entering to the traffic conditioning stage are already marked, they may get re-marked based on the TCS applied or the changing traffic conditions signalled by the metering operation.

### 4.3.1.4 Shaper

A shaper is responsible for delaying the traffic for transmission at specific times in order to match a particular line speed. Another purpose of employing a shaper is to reduce the burstiness of the traffic and match the flow pattern specified in the TCS. The shaper parameters are triggered by the metering operation and different shaping filters may apply to different QoS specifications. A shaper is made of a finite-size buffer and may discard packets if there is no sufficient space left to hold the delayed packets. The basic operation of a shaper is depicted in Figure 4.3.

### 4.3.1.5 Dropper/Policer

The dropping action is also known as policing of the traffic flows. A dropper simply discards packets in a traffic flow in order to match it with the specifications of the corresponding TCS. Based on its discarding algorithm, a dropper can be classified as an Absolute Dropper or an Algorithmic Dropper [BernY]. The effect of policing operation on the traffic rate is shown in Figure 4.4.
4.3.2 Forwarding Behaviours

After edge classification and conditioning, traffic flows are subject to differentiated forwarding treatments. In DiffServ architecture, each core router applies a predefined queueing and scheduling procedure on the incoming flows by identifying their DSCPs, and forwards them to the next hop (router). As defined earlier, a particular forwarding behaviour applied at a DS-node is named as PHB and the overall effect of PHBs on a behaviour aggregate inside the DiffServ domain is called the PDB. When a specific PHB or a group of PHBs and the edge conditioning mechanisms work together, they form the differentiated services. IETF has identified five forwarding behaviours and so far standardised four of them. These are:

- Default Behaviour
- Class Selector Behaviours
- Expedited Forwarding
- Assured Forwarding
- Lower Effort Forwarding
Chapter 4. Differentiated Services

QoS is realised based on dropping the “right” packets and assigning network resources for the transmission of others. Consequently, these forwarding behaviours are mainly distinguished with the way in which they implement traffic conditioning, queueing, scheduling and queue management mechanisms. In fact, DiffServ is not prescriptive in defining the scheduling and queueing control algorithms that should be implemented at each hop. Instead, it uses a level of abstraction in defining the externally observable behaviours (PHBs) that can be applied at each DiffServ compliant router (traffic hop). This approach is reasonable since the network operators may choose to configure their network services different from the others due to economical and technologic reasons. Nevertheless, there are number of effective tools that can be utilised to compose network services. The combinations of these tools create the difference in services. The above listed forwarding behaviours define the DSCPs and the traffic conditioning principles (not the rules) that are needed to be used for providing different ways of forwarding the network traffic.

4.3.2.1 Default Behaviour

Default Behaviour [NichK1] corresponds to the Internet’s default service where no guarantee about packet delivery is made; hence, packets are lost, reordered, duplicated and delayed at random. This type of forwarding treatment is suitable for non-real-time traffic (TCP flows). Ideally, network operators can provide a single queue with some kind of queue management mechanism to optimise the performance. The standardised DSCP value for default behaviour is zero.

4.3.2.2 Class Selector Behaviours

Class selector behaviours [NichK1] have been standardised to enable backwards compatibility with IP-Precedence approach. Seven PHB codepoints have been assigned corresponding to ToS field options on the IPv4 header. DSCP values run from 001000 to 111000, indicating a higher probability of timely forwarding as the numerical values of the codes increase. Class Selector compliant PHBs can be realized by a variety of mechanisms, including Strict Priority Queueing, Weighted Fair Queueing (WFQ), Weighted Round Robin (WRR), or variants [StilD], [BennJ], [ShreM], or Class-Based Queueing [FloyS1].

4.3.2.3 Expedited Forwarding

Expedited Forwarding (EF) was intended for providing a building block for low loss, low delay, and low jitter services [DaviB]. The details of exactly how to build such services are outside the scope of its specification. The aim of the EF PHB is to provide a PHB in which suitably marked packets usually encounter short or empty queues. Furthermore, by having short queues relative to the buffer space available, packet loss is also kept to a minimum. In EF, customers are expected to
produce traffic at fixed rate and the service provider guarantees to provide a service at or above a configured rate. EF was thought to be more suitable for real-time applications or high paying customers who would require a virtual leased line type network operation. Expedited forwarding requires a certain portion of network resources to be reserved for "premium" traffic at all times. This brings out concerns regarding the incremental deployment properties, complexity for network operators, additionally required router functionalities, and economic challenges.

The Internet2 consortium [Int2] QoS working group designed the QBone [Qbon] inter-domain DiffServ testbed to test their QBone Premium Service that was built on the EF principles. The main idea behind such service was to convert the packet switched network into a kind of wired network where promised performance guarantees could be provided at all times. The project was abandoned as it was decided that the deployment of a "premium" service had a very high cost relative to its perceived benefits. The detailed reasons about why Premium Service deployment was determined to be unfeasible is explained in [Arch]. Therefore, in this theses EF based approaches are not much elaborated. Instead, the focus was put on the Assured Forwarding and the Best-Effort approaches.

4.3.2.4 Assured Forwarding

Unlike EF, Assured Forwarding (AF) [HeinJ1] approach is designed to offer a service which cannot guarantee bandwidth but provides a high probability that AF marked packets get higher service reliability than default behaviour. In other words, AF service can be viewed as getting the standard QoS with a high change of getting more if there is no scarcity of network resources for any reason. The standard defines four different AF classes (AF1, AF2, AF3, and AF4). Each AF class is associated with different levels of QoS; that is to say, different levels of drop probabilities and assigned network resources. Within each AF class packets are marked with one of three possible drop precedence values. Therefore, it is possible to construct 12 different Assured Forwarding PHBs (AF11, AF12, AF13, AF21, AF22 ...AF43). The drop precedence of a packet is used to determine the relative importance of the packet within the AF class. In the case of congestion, the packets with lower drop precedence are subject to minor to moderate policing while others with higher precedence values receive stricter policing.

Packets in each forwarding class are queued and forwarded independently of each other. Meaning to say, flows with different AF markings are not aggregated together. DiffServ nodes reserve a certain amount of forwarding resources for each AF class (and the drop precedence). However, it is possible to reconfigure the distribution of these resources to each AFi (i = class, j = drop precedence) and assign more resources to higher priority AF classes in the case of no network congestion. However, AF classes are not associated with any quantifiable timing requirements (delay and jitter bounds). The AF implementation aims to minimise the long-term congestion

83
within each class by using active queue management techniques. The short-term congestion is tackled by queueing and dropping of excess packets.

If configured appropriately, it is also possible to service multimedia steams by using the Assured Forwarding approach (low loss and low delay). This may require an over-provisioned AF class and an admission control mechanism to control the maximum arrival rate of the input traffic flow. If low latency is the main requirement, no over-provisioning may be needed, yet the buffer space should have a low maximum limit for that particular AF class. The defining parameters of AF classes are specified in the SLA. Although static SLA is more common, particular attention should be paid on dynamic SLAs that can configure the AF queues and policing mechanisms associated with them. Assured service traffic is considered to be in-profile if customer traffic does not exceed the bit-rate specified by the SLS; otherwise, the excess packets are considered out-of-profile. Out-of-profile traffic is managed by the policing mechanism in charge.

An example service composition is presented in [HeinJ1] where simple differentiated service classes are formed by assigning different drop precedence to each AF class. This example is named as Olympic Service, which provides three tiers of service: gold, silver and bronze, with decreasing quality. The system uses a leaky bucket traffic policer that is parameterised by committed burst size and excess burst size. A packet is assigned low drop precedence if the number of tokens in the bucket is greater than the excess burst size, medium drop precedence if the number of tokens in the bucket is greater than zero, but equal or less than the excess burst size, and high drop precedence if the bucket is empty. The details of the traffic policing, queueing and scheduling mechanisms that can be used to implement various PHBs are given in Section 4.4.

### 4.3.2.5 Lower Effort Forwarding

Lower effort forwarding behaviour (LE) is defined for network traffic that has sufficiently low value and requires no explicit commitments for a QoS enabled delivery. It carries a lower precedence than the Best-Effort traffic. In other words, it is intended to consume the under-utilised resources of other traffic types. There may be little or no network resources reserved for this type of traffic, thus it may considered to be scavenging on reserved allocated for other traffic types [Qbss]. IETF follows the idea that no explicit PHBs are required to be defined for this type of forwarding behaviour, yet a PDB definition is published [BlesR] to provide the necessary details for the realisation of such service. This service can allow networks to protect themselves from certain types of traffic by not assigning any resources and preferential treatment. By this way, prioritised traffic aggregates may receive better service from the network as the possibility of congestion is minimised by using such forwarding mechanism. For instance, an excess UDP traffic may cause the depletion of network resources and cause congestion on the links. Instead of
strictly policing the excess traffic, LE forwarding behaviour may be assigned to ensure the optimal usage of resources and perhaps a "good faith" delivery of the excess traffic. Other possible applications that can make use of such degraded performance forwarding mechanism could be bulk emails, peer-to-peer file sharing traffic, web search engines traffic etc.

In this thesis, no further discussions regarding to the effects of using such service are presented. It is believed that LE forwarding is mainly useful for service providers who would like to have an alternative way of managing their customer traffic.

4.4 Traffic Control Mechanisms

In the previous sections, the fundamental principles and the components of the DiffServ architecture was explained. The framework for enabling the service differentiation was described. However, defining the traffic control mechanisms that are required for achieving the service differentiation is not a part of the DiffServ standard and the details of such implementation are left to the choice of the service operator. Such mechanism can be generalised into the following categories:

- Traffic Conditioning Algorithms
- Queueing and Scheduling
- Routing and MPLS

Although each of these mechanisms is a subject of extensive research, only the most popular ones and particularly those which are important for supporting real-time multimedia services are presented here. The research presented in this thesis concentrates on the AF approach. Therefore, the main focus is on the AF relevant traffic control mechanisms. Although there are numerous traffic control mechanism that could be utilised within a DiffServ model, only those that are suitable for demonstrating the benefits of differentiating services are presented.

4.4.1 Traffic Conditioning Algorithms

The traffic conditioning mechanisms are those, which are used to realise the functionalities within the marker, meter, shaper and policing blocks. Although these mechanisms are depicted as separate functional blocks, their operations are dependent on each other and it is possible to have the functionalities of one block embedded into the others (re-marking taking place at the policing). As shown earlier in Figure 4.2, the meter block is the heart of the traffic conditioning mechanism and is essential in terms of differentiating between the conforming and non-conforming flows. This is required for facilitating the operations of other blocks such as marking,
shaping, policing and queueing. The policing and shaping operations can as well be performed using simple token or leaky bucket algorithms. The policing operations or the shaping may exist as the part of the same block but it is not necessary to have both of them in a router if other mechanisms are configured to provide the required functionality.

### 4.4.1.1 Token/Leaky Bucket Algorithms

In [BernY], several possible meter algorithms were introduced amongst which token-bucket/leaky-bucket algorithms were more favoured [DaviB, HeinJl]. In general, token/leaky bucket algorithms are used for policing and shaping. These can also be utilised to determine the in-profile and out-profile flows (metering) and mark them for policing accordingly. While leaky bucket is realised with a finite first-in-first-out (FIFO) queue with a fixed scheduling rate, token-bucket algorithm defines a specific burst size that can be allowed before discarding any excess packets. Figure 4.5 shows the simple Token Bucket (TB) principle.

![Fig 4.5: The Simple Token Bucket Operation](image)

A meter monitors the input traffic streams and sends control messages to other traffic conditioning elements to dynamically modulate their behaviour based on the conformance of a packet to a predefined specification (TCS). This specification includes meter parameters that specify the temporal profile and conformance levels of a specific class of flow. A meter can be used to trigger real-time traffic conditioning actions by routing a non-conforming packet through an appropriate next-stage action element. Conformance can be measured to a token bucket profile. A TB profile generally has two parameters.

- \( R \) = Average Token Rate
- \( b \) = Maximum Allowed Burst (bucket) Size (bytes)
TB meters compare the arrival of packets to the average rate specified by the TB profile. Tokens accumulate in the bucket at an average rate of $R$. Token rate can be related to bytes/second. The relationship between $R$ and $b$ can be defined as:

$$R \times t = b \tag{4.1}$$

For instance, say information rate (token rate) is $R = 1.2$Mbps and the burst size ($b$) is 1500 bytes. In this case, $t = 10$ms, meaning that the conforming traffic will arrive (in the worst case) 100 bursts per second of 1500 bytes each at an average rate not exceeding 1.2Mbps. A data stream is said to conform to a simple token bucket parameterised by a $\{R, b\}$ if the system receives at any time interval, $t$, at most, an amount of data not exceeding $(R \times t) + b$.

Similarly, multi-rate token parameter mechanisms can be used to measure the conformance to more than one specified rate. The marker may make use of this conformance information to mark/re-mark the packets for further treatment in other traffic conditioning or queueing components.

For the assured forwarding service where different drop precedence is assigned to different traffic flows a multi-rate meter/marker can be used. Two traffic meter/marker combinations were proposed for AF PHBs in [HeinJ2, HeinJ3] that use token buckets to meter the streams in terms of their burst sizes and information rates, and colour mark them based on the conformance levels. In these, three levels of conformance are discussed which are represented by colours. Green represents conforming, yellow partially conforming and red non-conforming.

The “Two Rate Three Colour Marker” (trTCM) [HeinJ3] meters an IP packet stream and colour codes the packets accordingly (a specific DSCP is coded into the IP header). This marker uses four parameters: the Peak Information Rate (PIR), the Committed Information Rate (CIR) and their associated burst sizes — a Committed Burst Size (CBS) and an Excess Burst Size (EBS). The burst sizes are usually given in bytes, the rates in bits per second (bits/s). A packet is marked red if it exceeds the Peak Information Rate (PIR). Otherwise, it is marked either yellow or green depending on whether it exceeds or does not exceed the committed information Rate CIR. This marker is useful for ingress policing of a service, where a peak rate needs to be enforced separately from a committed rate. This type of marker is usually implemented using two token bucket filters with PIR and CIR as bucket rates, CBS and EBS as bucket sizes. The implementation of this marker is shown in Figure 4.6.
The other marker proposed is the “Single Rate Three Color Marker” [HeinJ2]. In contrast to the “Two Rate Three Color Marker” only one rate and two burst sizes are specified. A stream is metered and marked according to three traffic parameters Committed Information Rate (CIR), Committed Burst Size (CBS), and Excess Burst Size (EBS) to be green, yellow, or red. A packet is marked green if it doesn’t exceed CBS, yellow if it exceeds CBS but not EBS, and red otherwise. This marking mechanism is useful for ingress policing of a service, where only the length but not the peak rate of the burst determines service eligibility.

![Implementation of Two Rate Three Colour Marker](image)

Figure 4.6: Implementation of Two Rate Three Colour Marker [HeinJ3]

TB algorithms have certain drawbacks. This is because the TB meter is theoretically designed for fixed-length units of data. That is to say, a TB mechanism assumes a leaky bucket traffic shaping operation had taken place in the previous hop. Most of the existing applications produce variable-length packets, and usually the user makes little or no attempt to shape their output stream. The variance in packet sizes may cause jitter in the packet stream. Both token buckets presented here are subject to a persistent under-run. These accumulate tokens over the time; up to the maximum burst size. If the maximum burst-size is exactly the size of the packets being sent, the token depth becomes zero and starts to accumulate again. If the next packet is received any time earlier than a token interval later, it will not be accepted. Conversely, if the next packet arrives exactly on time, it will be accepted and the token depth again set to zero. If it arrives later, however, accumulation of tokens will have stopped because it is limited by the maximum burst size: during the interval between the bucket becoming full and the actual arrival of the packet, no new tokens are added. As a result, jitter that accumulates across multiple hops in the network conspires against the algorithm to reduce the actual acceptance rate. Therefore, it is a sensible approach to set the maximum token bucket size somewhat greater than the maximum possible Media Transmission Unit (MTU) size in order to absorb some of the jitter and allow a practical acceptance rate more in line with the desired theoretical rate.
4.4.1.2 Absolute and Algorithmic Droppers

An absolute dropper is an entity, which discards the chosen packets and does not forward them to the queueing processes. Its operation is simple and does not take any parameters. It only checks the packet marking and either destroys or forwards it. This is commonly used to discard the out-of-profile or unauthorised traffic entering the Diffuser domain. The absolute dropper could be considered as a part of the policing block, but not the only mechanism that could discard packets in a DiffServ border router. That is to say, if a TB mechanism is in place for traffic shaping operation, it could as well discard some of the packets. Nevertheless, such TB mechanisms discard packets based on some conformance levels with no consideration to packet markings.

Algorithmic droppers are different from absolute droppers. They can take parameters and their behaviour can be modified to produce different discarding algorithms. One of the primary characteristics of an algorithmic dropper is the ability to choose which packet to discard. It is usually considered as a queue management mechanism that can selectively drop packets within a queue. In its typical implementation, an algorithmic dropper would be connected to the input of a FIFO queue, monitoring the queue depth over a certain period and providing feedback to its discarding mechanism. Figure 4.7 shows an example of how an algorithmic dropper would work if attached to the end of a FIFO queue.

Algorithmic droppers can be utilised to differentiate between different QoS classes. They manage the length and the contents of the queue by selectively discarding the lower priority packets. It is also possible to configure them for random dropping of the TCP packets. This triggers the TCP flow control mechanism at different end-hosts and reduces send rates at different times. By this way, the likelihood of having congestion collapse is reduced and the network utilisation does not oscillate [XipeX], [FloydS3]. Further details on algorithmic droppers and their importance for AF services are explained in the next section.
4.4.2 Queueing and Scheduling

Having classified and conditioned the user traffic streams based on the agreed specifications, what required is the differential management and forwarding of the traffic aggregates. This treatment should ensure minimal congestion in the network and provide a hierarchical distribution of the network resources. To handle packets differently, a DiffServ router should contain a queueing system installed to its outgoing interfaces. A queueing system is a data network where packets arrive, wait in various queues, receive service, and exit after some time. While a queueing system could be as simple as a single FIFO queue, DiffServ routers usually need to employ mechanisms that are more sophisticated. This allows the handling of the each traffic class differently by putting them into different kind of queues and processing these queues with differential priority. Although there are multitude of different queueing and scheduling mechanism in the literature, only the most common disciplines that can be utilised to achieve the required type of service differentiation are presented here. In [BakeF], queueing systems are divided into two main categories as priority queueing and rate queueing. A priority queueing system is composed of a set of queues and a scheduler that serves them in priority cycle. Priority queueing (PQ) schemes are used to guarantee specific delay, variations in delay, and packet loss in the network. Such schemes are usually associated with EF service, where a certain portion of network resources is reserved for specific flows. Conversely, rate-based queueing schemes are generally associated with AF service classes and the scheduler services every queue in the system at a specified rate, hence providing opportunity for all aggregates to receive a level of service. Common examples to rate a queueing system are Weighted-Fair Queueing (WFQ), Weighted Round-Robin (WRR), and Deficit Weighted Round-Robin (DWRR). In a typical DiffServ router, it is expected to find both
types of queueing systems, e.g. a PQ for EF class, a WFQ for AF classes, and a rate-limited FIFO for others.

Queue scheduling disciplines are effective when the system is stable (i.e., the packet arrival rate is less than the system transmission capacity). If congestion occurs, there is not much benefit in employing advanced queue scheduling disciplines for different types of traffic aggregates. As a result, there is need for queue size management to be able to bias the traffic flux and prevent the queue overflows, which can have adverse effects on the application QoS performance [BradB]. Therefore, the Active Queue Management concept is presented first before introducing the queue scheduling disciplines.

**4.4.2.1 Active Queue Management**

Queue management is used for controlling the number of packets in a queue by determining when and which packets should be dropped. This is commonly performed by monitoring the occupancy level of a queue and removing certain packets from it based on an algorithm. Traditionally, queues are set to have a maximum limit beyond which all of the subsequently arriving packets are dropped (tail-drop effect). However, this simple management mechanism has two significant disadvantages. Firstly, it has no mechanism to differentiate between different flows, hence it is likely that several aggressive flows consume all of the queue space, causing other connections to starve for bandwidth. As a result, if the queue is full, an arriving traffic burst will experience multiple drops. TCP flows that are subject to this effect experience synchronised throttling back of the flows, followed by a sustained period of lowered link utilization and reduction in overall throughput. Ideally, the queue occupancy should be kept low so that the traffic bursts can be accommodated as the applications require, and the end-to-end delay is kept low while maintaining a high throughput.

The solution to these problems is to detect and drop the packets before the congestion occurs. This helps the congestion avoidance mechanism of TCP flows to reduce the sending rate, consequently reducing the likelihood of network congestion. Keeping the queue size within certain limits can help to regulate which flows take how much of the queue resources and achieve lower end-to-end delay, higher throughput and better link utilisation. This proactive queue management approach is called *active queue management* (AQM) [BradB]. Congestion and flow control mechanisms based on this approach are widely advocated [FloyS2], [ParrM].

In essence, AQM is an algorithmic dropper mechanism. The traditional method used for AQM is the Random Early Detection (RED) [FloyS3]. RED monitors the load at the router queues and stochastically discards packets when congestion is foreseen. The RED mechanism discards the
incoming packets with a dynamically computed drop probability. By dropping some of the packets before a queue is full, the average queue size is kept low and dropping large numbers of packets at once is avoided. RED is the basic method that reduces the chances of tail drop and allows better utilisation of the network links. However, it was particularly designed to work with responsive flows that are transported using a reliable mechanism (i.e. TCP). Real-time applications also benefit from RED since it helps to keep the average queue sizes short. Consequently, the end-to-end delay experienced by packets is reduced. Nevertheless, real-time applications are carried over User Datagram Protocol (UDP) and have a limit to packet loss before the application becomes unusable. As a result, if RED is to be used for UDP flows, its parameters should be carefully adjusted.

RED makes use of drop profile (see Figure 4.8) to control the aggressiveness of its packet drop process. This profile describes a range of drop probabilities across a range of queue occupancy states. A simple RED mechanism has four parameters:

- $th_{\text{min}}$: minimum threshold
- $th_{\text{max}}$: maximum threshold
- $w_q$: queue weight
- $max_p$: maximum drop probability

For every arriving packet, the RED algorithm needs to compute the average queue size, which determines the degree of burstiness that is allowed. The average queue length is calculated by:

$$avg = (1 - w_q) \cdot avg + q \cdot w_q$$  \hspace{1cm} (4.2)
where \( q \) is the actual queue size and \( \text{avg}' \) is the old average value. From the drop profile figure, the initial packet drop probability can be defined as:

\[
P_b = \max_p \left( \frac{\text{avg} - \text{th}_{\min}}{\text{th}_{\max} - \text{th}_{\min}} \right)
\]  

(4.3)

The dropping probability linearly increases from 0 to \( \max_p \). The actual drop probability makes use of \( p_b \) and the number of packets since last dropped packet. This is given as:

\[
P_a = \frac{p_b}{1 - \text{count} \times p_b}
\]  

(4.4)

where \( \text{count} \) corresponds to number of counted packets since the last dropped packet. In this system, it is important to determine how fast the average queue length follows the value of the actual queue length \( q \). The \( w_q \) parameter is used to control how fast RED reacts to the bandwidth fluctuations. The details of how to select an appropriate value for \( w_q \) are given in [FloyS3]. There, it was suggested that \( w_q = 0.002 \) is a suitable choice. The other challenge in designing RED is how to set the \( \text{th}_{\min} \) and the \( \text{th}_{\max} \) values. The optimal values for these parameters depend on the average queue size and the characteristics of the typical traffic. For instance, if the traffic is bursty, \( \text{th}_{\min} \) should be set to be high enough to account for this burstiness, hence prevent possible low link utilisation. Thus, \( \text{th}_{\min} \) should be larger than the largest possible burst size. Once this value is determined, \( \text{th}_{\max} \) should be set as at least twice the value of \( \text{th}_{\min} \). This corresponds to the typical increase in the calculated average queue size in one round-trip time. Given these, a RED algorithm is a function with variables \( \text{th}_{\min}, \text{th}_{\max} \), and \( \max_p \), and can be represented as:

\[
Q_{\text{RED}} = (\text{th}_{\min}, \text{th}_{\max}, \max_p)
\]  

(4.5)

When a packet arrives at the RED managed queue, the new average queue size is calculated (\( \text{avg} \)) first. If the \( \text{avg} \) is smaller than \( \text{th}_{\min} \), then the packet is stored in the queue. If \( \text{avg} \) is larger than \( \text{th}_{\max} \), the packet is dropped. If the average queue size resides between the thresholds \( \text{th}_{\min} \) and \( \text{th}_{\max} \) the arriving packet is likely to be dropped with a probability of \( p_a \) or queued with a probability of \( 1 - p_a \). As can be seen from Equation 4.3 and Equation 4.4, the drop probability increases linearly with the increasing queue size.

In [HeinJ1], a RED based mechanism was suggested for AF service. In [BodyU], this fact was highlighted and a RED variant algorithm to achieve load tolerant service differentiation was proposed. As described earlier, an AF class has three levels of drop precedence, which require differential traffic conditioning and forwarding treatment. A typical AF Meter/Marker mechanism
explained in Section 4.4.1.1 provides three levels of differential marking amongst the traffic of the same AF class. Conversely, end-user may provide the marking of their applications. Usually, it is preferable to place all three types of marked packets into the same queue to prevent the reordering of packets. Therefore, the AQM mechanism for such queue should be able to apply different packet handling policies. That is to say, a different set of $Q_{RED}$ should be applied to each flow within the same queue to achieve preferential drop in the case of congestion. If the three colour marked packets can be categorised as having low, medium and high drop precedence, the following RED parameter set should be applied:

$$Q_{RED} = (th_{min}^L, th_{max}^L, \max^L_p)$$

$$Q_{RED} = (th_{min}^M, th_{max}^M, \max^M_p)$$

$$Q_{RED} = (th_{min}^H, th_{max}^H, \max^H_p)$$

(4.6)

The drop profile for three RED algorithms for the same queue is depicted in Figure 4.9. This configuration is also called the Weighted-RED (WRED) [Cisc].

In summary, AQM helps the DiffServ architecture to enable service differentiation at the packet level. RED is the most widely used AQM technique. Number of RED variants has been developed by the research community. Amongst those, RIO [ClarD], WRED [Cisc], BLUE [FengW], FRED [LinD], REM [AthuS], SRED [OttT], RED-PD [MahalR] are the better recognised ones. However, RED is the most popular one and is widely deployed in the Internet.
4.4.2.2 Queue Scheduling Disciplines

Queue scheduling disciplines are used to control access to output link bandwidth by methodically selecting the next packet to be transmitted. The selection of which scheduling discipline to use is an important choice as it affects the delay, jitter and the packet loss experienced by the end-users. An effective scheduling discipline is required to accomplish the following:

- Provide a fair distribution of the bandwidth between different service classes competing for the same output link.
- Prevent certain flows from adversely affecting the performance of others.
- Enable a soft priority allocation between different classes so that the unused resources of one service class can be assigned to another.

**First in First out (FIFO)**

The FIFO queueing is the most basic queueing discipline. All the packets in a FIFO queue are treated equally and serviced at the same order that they were placed in the queue. In the FIFO example, there is a single finite length buffer with a single server scheduled to pull packets and push them into the output port at regular intervals. The packets in a FIFO queue are not reordered and are transmitted at exactly the same sequence that they arrive. The delay experienced by the traversing traffic is proportional to the queue length. However, FIFO is not suitable for providing class based differential treatment of the packets. Because of this, it fails to protect sensitive applications during the times of congestion; thus, results in increased delay, jitter and loss for applications. Unless strict admission and flow control mechanism are in place, it is likely that unresponsive UDP flows may take over all the buffer space, causing the TCP applications to throttle down or eventually creating denial of service for other users [FloyS2].

**Priority Queueing (PQ)**

PQ refers to a class of queue scheduling algorithms that are designed to provide simple method of supporting differentiated servicing of different traffic aggregates. In its essence, traffic flows are aggregated into different priority queues. The scheduler prefers queues with high priority; forwarding packets from queues with lower priority only when the higher priority queues are empty. There may be a number of queues with varying priorities. Generally, within each priority queue, packets are scheduled in FIFO order. The scheduler scans the queues from their highest to lowest priority, looking for any packet to transmit. Once a packet has been sent, the scheduler restarts scanning from the highest priority (see Figure 4.10). Therefore, the packets placed into the queue with the highest priority experiences the best possible service, while packets in other queues with lower priority can only be serviced when the highest priority queue is empty.
The main use of PQ is to allow a particular type of traffic to get a reserved portion of the resources, hence treated differently from other traffic classes. This discipline is designed especially for time sensitive data, such as interactive voice and video applications or mission critical administrative or network control traffic.

As long as high priority packets do not exceed the outgoing bandwidth, the packets are forwarded with a minimal delay, jitter and loss (EF forwarding in DiffServ). However, implementation of PQ requires the expedited forwarding traffic to be policed as misbehaving traffic is able to completely starve the bandwidth of traffic classes with less priority. PQ does not serve as a solution to the UDP/TCP fairness problem highlighted in the previous section. If UDP flows are prioritised, they tend to consume all of the resources and if TCP flows are favoured, then the TCP flow control mechanism will keep increasing the sending rate and starve the lower priority UDP flows.

Usually, PQ takes two forms. The first one is strict PQ and the other one is the rate-controlled PQ. In strict PQ, there are no limitations on how much of the network resources can high priority flows consume. Simply, high priority flows are allowed to starve other flows if they require. On the contrary, rate-controlled PQ allows high priority flows to be scheduled before the lower ones only if the amount of traffic in the higher priority queue stays within predefined bounds (e.g. higher priority traffic is not to consume more than 20% of the output link bandwidth). A useful PQ implementation demands that the traffic is conditioned at the edges of the DiffServ network to prevent higher priority queues from becoming oversubscribed.
Chapter 4. Differentiated Services

Fair Queueing (FQ)

The principle of FQ [NagJ] is to allow each traffic aggregate to have a fair share of the available network resources and prevent the bursty flows from consuming all the network resources by starving the others. In FQ, each traffic aggregate is assigned a separate queue and serviced as often as every other queue. Each queue is serviced one packet at a time in a round-robin order. The advantage of FQ is that it assigns the same priority to both UDP and TCP flows, hence, in theory, preventing them from consuming more than what is their assigned fair share of the network resources. If a particular traffic aggregate tends to become more aggressive, the only queue that is affected would be its own one (queue may overflow), making no effect on the performance of the other queues on the same shared output link. In fact, FQ overlooks the fact that different applications have different bandwidth requirements as well as delay, loss, and jitter constraints. On the other hand, FQ provides equal share of the available bandwidth only if the sizes of the packets in each queue are the same (servicing one packet per queue per time); otherwise, applications with larger packet sizes receive a larger share.

A variant of FQ is class-based FQ, where the queues are divided into several classes. Each class is assigned a certain portion of the available bandwidth. Then, every queue in a particular service class is serviced with a dedicated FQ server, which is independent of other FQ servers managing the rest of the queue classes. Consequently, all the flows assigned to a given service class are provided equal shares of the aggregate bandwidth configured for that specific service class [Juni1].

Weighted Fair Queueing (WFQ)

The DiffServ requires certain flows to receive preferential treatment over the others. The FQ algorithm, although can fairly share the available resources between different kinds of traffic, fails to recognize the fact that not every traffic aggregate is required to receive the same amount of available bandwidth. WFQ overcomes this by assigning each queue a weight, which determines the queue servicing frequency [DemeA], [ZhanL]. For instance, if a particular queue is assigned a weight of 50%, this means that the scheduler needs to service this queue 50% of the time, in other words half of the output bandwidth is assigned to it. However, schedulers dequeue one packet per service instant from a queue. Therefore, if the average packet sizes of traffic aggregates are not equal, fair distribution of the bandwidth between queues based on their assigned weights cannot be achieved. WFQ supports the fair distribution of bandwidth for variable packet sized aggregates by approximating a Generalized Processor Sharing (GPS) system. The GPS behaviour can be represented by a weighted bit-by-bit round-robin scheduler. In theory, such scheduler removes a single bit from the head of a queue every time it needs to service it. WFQ approximates this
Chapter 4. Differentiated Services

theoretical scheduling concept by calculating a variable called the Virtual Finish Time (VFT). For every queue, a VFT is calculated for each arriving packet to the queue and the scheduler chooses the one with the smallest value. Every new packet arrival initiates a new VFT calculation, which takes into account the instantaneous activity level of other queues as well as the weight of the queue. This value is tagged onto the packet. The VFT is calculated using the following formula [DemeA], [ZhanL]:

\[
VFT_a(i,k,t) = \max\{VFT_a(i,k-1,t), R(t)\} + \frac{P(i,k,t)}{\omega_i}
\]

where,

\( VFT_a(i,k,t) \): VFT for packet \( k \) on arriving into queue \( i \) with weight \( \omega \) at time \( t \).

\( P(i,k,t) \): Size of packet \( k \) in queue \( i \) arriving at time \( t \).

\( R(t) \): Round number at time \( t \) that represents the number of rounds completed for active flows.

\( \omega_i \): Weight given to queue \( i \).

If any of the queues become inactive, the scheduler service only the active queues, hence increasing the share of bandwidth they receive. The weighted servicing of queues brings an upper bound to the rate at which every aggregate can receive. This prevents UDP flows from consuming all of the network resources. Therefore, WFQ can be seen as another traffic conditioning mechanism in a DiffServ router. On the other hand, if WFQ scheduling is used together with token or leaky bucket rate control algorithms, it can provide end-to-end delay performance guarantees [PareA]. A simple illustration of WFQ scheduling with VFT tagged packets is shown in Figure 4.11.

Nevertheless, WFQ has certain drawbacks. That is to say, WFQ scheduling has a complex mechanism and requires computations to take place for every flow and packet. In that sense, it is a CPU consuming process and may not be suitable for implementation on high-speed links. Because of this complexity, the network operator may tend to choose coarser granularity of flow aggregation, i.e. less number of queues, which in turn may result in degradation in the QoS seen by the flows within the same aggregate.
WFQ is one of the most popular queueing methods that can provide a fair servicing discipline to the traffic aggregates. It has been widely recognised by both the industry and the research community and is already embedded in many commercial network routers. WFQ was designed before the DiffServ concept with the IP precedence approach in mind. Over the years, a number of different versions of WFQ have been developed to overcome its limitations. Among these, worst-case fair weighted fair queueing WF^2Q [BennJ2] and the WF^2Q+ [BennJ] are the versions that have improved the fairness and decreased the complexity of WFQ.

**Weighted Round Robin (WRR)**

Weighted Round Robin scheduling (also known as Class Based Queueing (CBQ)) was designed to address the limitations of FQ and PQ scheduling mechanisms [HahnE]. Its operation resembles to WFQ in the sense that it divides the available bandwidth between different traffic aggregates. WRR incorporates multiple FIFO queues with a specifically assigned weight that defines the percentage of the link resources allocated to it. Unlike WFQ, with WRR, at least one packet is removed from each queue during each service round. Packets are first classified into different service classes and then inserted into a particular queue that is dedicated for that service class. In an environment where bandwidth should be shared proportionally between different traffic flows, WRR provides a flexible and efficient mechanism since it does not involve VFT calculation for every arriving packet.

Since WRR services at least one packet per queue for every round, it cannot provide an accurate share of bandwidth (i.e., it is not fair unless the packet sizes are all equal). Therefore, it can only offer a coarse allocation of bandwidth amongst the traffic aggregates. To regulate the allocation of resources to each service class for the desired performance requires a careful tuning of a number
of parameters, i.e. queue weights and depths given an output link speed and the number of other queues to be served. Other than its simplicity, the basic advantages of the WRR discipline are that all traffic aggregates have access to at least some configured amount of network bandwidth and it provides an effective servicing of the differentiated service classes to a reasonable number of aggregated traffic flows (coarse aggregation).

**Deficit Weighted Round Robin (DWRR)**

Deficit Weighted Round Robin scheduling discipline (DWRR) [ShreM] is similar to WFQ and WRR but was designed to overcome the shortcomings these models. As mentioned earlier, WRR scheduling fails to provide a fair service to traffic aggregates with variable-length packets. On the other hand, WFQ scheduling is complex and is difficult to implement on high-speed links. The DWRR model has lower complexity than WFQ and is capable of providing service to queues that contain variable-length packets.

Three parameters characterise the operation of a DWRR scheduler. The *weight* indicates the share of output link bandwidth assigned to a particular queue. The total number of bytes that a queue is permitted to transmit is defined by a parameter called the *deficit-counter*. The *quantum* parameter is related to the weight of the queue and is expressed in terms of bytes. Every time the scheduler services a queue, the deficit-counter is incremented by that queue’s quantum number, and the scheduler forwards deficit-counter number of bytes from that queue.

The DWRR scheduler keeps a list of active (non-empty) queues in its system and services them in round robin order. If any queue is empty, its corresponding deficit counter is set to zero. For every active queue, the size of the packet at the head of the queue is determined. At this moment, the deficit-counter is incremented by that queue’s quantum parameter. The size of the quantum parameter is determined in proportion to its weight. For instance, if a particular queue’s weight is 50% and its quantum is determined to be 100 bytes, another queue with a weight of 25% would have a quantum equal to 50 bytes. If the packet size is greater than the deficit counter, then the scheduler moves on to the next active queue but does not reset the value of the deficit counter. A packet or a number of packets are removed from the queue as long as the deficit number is equal or larger than the size of the packet at the head of the queue. When a packet is removed from a queue, its deficit counter is decremented by the removed packet size amount. Once a queue is emptied, the deficit counter is reset to zero and gets incremented again only when the queue becomes active again. The list of the active queues is constantly updated every time a new packet arrives in the system.
Chapter 4. Differentiated Services

DWRR scheduling has a number of advantages. Firstly, it is an effective way of isolating flows of different characteristics, hence preventing the unresponsive aggregates from consuming all of the bandwidth and starving other aggregates. Secondly, the deficit counter mechanism makes this discipline fair for aggregates with variable-length packets. Moreover, it has a simple algorithm, which is well suited for implementation on high-speed links. The main disadvantage of the DWRR is that it cannot provide strict delay guarantees as WFQ.

4.4.3 A Practical Implementation

In previous sections, traffic conditioning, queue management, and scheduling algorithms have been introduced. The DiffServ architecture does not define which algorithm or combinations of algorithms should be used to support its forwarding services (e.g., EF and AF). However, the analysis of these behaviours yields an indication to which algorithms could be utilised to achieve the required functionality. For instance, it is clear that a weighted scheduling discipline together with a kind of algorithmic dropper mechanism is useful for the AF service. On the other hand, EF service requires priority handling with certain performance guarantees. This prioritised handling of the EF behaviour should be implemented without significantly hindering their performance assurances. In the light of these facts, it could be argued that there cannot be a single model that can be classified as a generic DiffServ node model that favours the implementation of specific algorithms. For this reason, IETF has provided certain guidelines indicating the possible functionalities that a DiffServ router should possess [BakeF2] [BernY].

Nevertheless, having introduced the theoretical background in the previous sections, a practical DiffServ node model can be devised. The model shown in Figure 4.12 represents a practical DiffServ edge router implementation. This node contains all of the fundamental elements for providing differentiated services to the traffic flows. Usually, only EF, AF and BE PHBs are supported in the DiffServ domains. Thus, the model depicts six separate queues; one for the EF service, four for the AF service and another one for the BE service. DWRR was chosen as the scheduler for different AF classes. Commercial router manufacturers such as Cisco and Juniper Networks also make use of WDRR or its variants in their DiffServ capable products.

All three forwarding behaviours are served with a rate-controlled PQ scheduler which a gives the highest priority to the EF aggregate, the middle priority to the AF aggregates and the lowest priority to the BE traffic. The output of the EF queue is shaped by a TB mechanism to ensure that the performances of other services are not adversely affected by instantaneous fluctuations of its traffic. Each AF queue is policed with either an AQM or a TB mechanism. At the ingress, the
flows are classified and marked into different aggregate classes. Metering, remarking and dropping mechanism are applied to ensure TCS conformance.

![A Practical DiffServ Edge Node Model](image)

**Figure 4.12: A Practical DiffServ Edge Node Model**

### 4.4.4 Routing and MPLS

The DiffServ standard is mainly concerned with the traffic conditioning, aggregation, and the forwarding behaviours, yet makes no explicit reference to any routing protocols. However, it is clear that a means of a differentiated forwarding mechanism can be useful in achieving end-to-end service guarantees and help the network operator provide the promised service assurances stated in SLAs. The standard routing protocols are mainly based on shortest-path algorithms and do not incorporate any mechanisms to recognise the DSCP mark on the IP headers. Consequently, all the packets are forwarded with the same algorithm with no consideration to their QoS requirements. As a result, any routing mechanism that is required to supplement the DiffServ architecture should be able to route different flows based on QoS requirements. For instance, a less congested but a longer path could provide a better packet loss and delay performance compared to the shortest path. The benefits of providing alternative routing paths for the requiring traffic aggregates become more obvious in large networks where there may be many routers between the end-users.

Traffic engineering is the process of how traffic aggregates are routed in the network so that the routing protocol’s adverse effects, i.e. uneven link utilization and congestion, are alleviated. MPLS is a Layer-2 protocol used for traffic engineering, hence can improve IP routing efficiency.
Chapter 4. Differentiated Services

[AwduD]. It can perform an even distribution of the traffic load between the network links as well as creating a virtual tunnel that enables fast routing of the label-tagged traffic streams. Therefore, the combination of DiffServ and MPLS presents a very attractive strategy to network providers.

Multi-Protocol Label Switching (MPLS) [RoseE] is a mechanism that provides bandwidth management for aggregates through network routing control according to the labels on packet headers. In MPLS, the QoS and routing related information on layer-3 packet header (IP header) are mapped to connection-oriented Layer-2 transport mechanisms like ATM and Frame Relay. MPLS makes use of a label tagged on the packet headers, which contains specific routing information for each IP packet and allows routers to assign explicit paths to various classes of traffic.

Both DiffServ and MPLS are domain-based architectures and show similarities in the way they structure their functionalities. Like DiffServ, MPLS also pushes the complexity to the edge of the network. The density of traffic that the edge routers need to deal with is comparatively less than those routers in the core network. This allows for the classification of the traffic streams without being overwhelmed with the traffic traversing them. Having performed the necessary classifications, the packets are tagged on labels indicating the forwarding treatment they require. Routers in the MPLS domain's core treat the packets according to their tags. This is analogous to the per-hop behaviours in the DiffServ domain where the DSCP marking on the IP headers determine the behaviour of the nodes in terms of scheduling mechanisms and queueing management. The MPLS label of a packet determines the path the packet takes and the packet is routed based on its label. Traffic engineering is performed by assigning certain labels to paths with certain characteristics. The combination of features from both MPLS and DiffServ [FaucF] would result in increased efficiency of QoS provisioning in the network. In order to use DiffServ in MPLS networks, the DiffServ related information contained in the IP header should be mapped to the MPLS label assigned to the packet.

This mapping can be done in several ways. The network administrator has to decide how the DSCPs are mapped into the MPLS LSPs (Label Switched Path). If multiple DSCPs are mapped into the same LSP, all the packets will be treated the same way by the LSRs (Label Switched Routers). The experimental field in the MPLS header is used to specify the PHB applicable to each packet. The PHB includes scheduling and aggregate handling instructions. This way of mapping is referred as E-LSP (Exp-inferred).

If the mapping is done on one-to-one basis, the DSCP is encoded implicitly in the label. Therefore, the experimental field can be utilised to indicate the aggregate handling, i.e. AQM parameters, instructions. This is known as L-LSP (label only inferred). L-LSP provides a more
QoS efficient way of mapping, but it imposes higher requirements on the system. As the number of labels increase more resources would be needed in the network. This may lead to scalability problems. Maintaining a high amount of labels can become a problem when the number of PHBs in the system increases. For the traffic streams that do not demand prioritised treatment, there is a best effort label for any LSP.

A LSP with special PHB can be created dynamically upon a request. The need for a LSP with such PHB needs to be signalled via a reservation protocol like RSVP. When a request is received, the MPLS network creates a label with the required characteristics and defines its possible path. This method reduces the number of labels to be maintained but introduces more traffic and delay in the connection due to the signalling overhead.

In order to prevent this signalling protocol overhead, a traffic driven label distribution approach can be used. In this case, labels are established upon reception of data traffic. By this way, the number of labels can be kept at minimum. However, in this approach, the initial packets of flows cannot be provided with the QoS desired since the proper labels are in the process of establishment. Therefore, the beginning of every flow will not conform to the SLS between the user and the network. This disadvantage may affect different applications in different way, yet the effect can be considered negligible if the flow is held for a long time. Nevertheless, networks handling short duration flows cannot use this data traffic driven approach.

In summary, MPLS and DiffServ can be regarded as complementary solutions for the problem of providing different service levels in a single network domain. DiffServ provides different traffic aggregate conditioning and forwarding in the nodes while MPLS deals with the paths between different nodes. In this research, the possible advantages of MPLS were not investigated, as the scope was limited to network, transport and application layer strategies for QoS provision.

### 4.5 Traffic Categories and Service Classes

A functional DiffServ architecture requires the service provider to categorise the various types of traffic that needs to be serviced in the network. This categorisation should consider the requirements of the applications, the users, and the organisation/customer who is footing the bill for the QoS services. For instance, an application should be able to receive better treatment, i.e. guaranteed service, if the customer is willing to pay for it. Nevertheless, a basic classification of traffic can be done by considering the application requirements. In [G1010], the user traffic is
broadly classified into four categories. These are namely, interactive, responsive, timely, and non-critical. These traffic types have varying delay requirements as shown in the Table 4.1.

The network control traffic is a separate category than the user traffic types. It includes the information that is essential for the operation of the administered network domain.

<table>
<thead>
<tr>
<th>User Traffic Types</th>
<th>Delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interactive</td>
<td>&lt;&lt; 1s</td>
</tr>
<tr>
<td>Responsive</td>
<td>~ 2s</td>
</tr>
<tr>
<td>Timely</td>
<td>~ 10s</td>
</tr>
<tr>
<td>Non-critical</td>
<td>&gt;&gt; 10s</td>
</tr>
</tbody>
</table>

Table 4.1: Delay Requirements of Different User Traffic Types [G1010]

The signalling used between different domains, routers and servers are also classified as network control traffic. The network operator may choose to treat these kinds of flows preferentially. For network maintenance reasons, it is important to sustain the flow of the control traffic, even if the network is congested. The preferential treatment of network control traffic is not investigated in this research. However, realisation of how different applications should be treated in consideration to their delay requirements is important for enabling QoS in the DiffServ architecture.

The four traffic types mentioned above can further be categorised into one or more different service classes. End-to-end performance requirements for these traffic categories are defined in [G1010] and [Y1541]. Service classes are used to identify the necessary treatment for the traffic in order to meet the user, application and the network requirements. In accordance with [BakeF], seven service classes can be defined to represent the user traffic. These service classes and example application types are presented in Table 4.2.

A typical network is not required to provide support for all service classes. Instead, the network administrator needs to choose the supported service classes based on their needs, i.e. customer requirements and application types in use. It is also possible, if required, to aggregate several service classes into a single behaviour aggregate. However, a SLA should be defined and enforced for every service class or aggregated service classes. In the following subsections, these service classes are further discussed in relevance to their handling in a DiffServ domain.
4.5.1 Telephony

The Telephony service class is for real-time applications that have very stringent delay and packet loss requirements. Typically, the service offered is a higher priority scheme where traffic belonging to this class receives a reserved portion of network resources and is subject to minimal delay and packet loss. In a DiffServ network, the telephony service class is realised by using EF PHB. In general, the application traffic packets have fixed sizes, which are emitted from their source at a constant rate. In addition, the traffic demanding to use this service class needs to be admission controlled to ensure that the admitted traffic will be within the defined limits of the domain’s EF resources. This admission control can be performed by gatekeepers (H.323) or call servers using signalling at access points to the network. As shown in Figure 4.12, the traffic aggregate of the telephony service needs to be policed to ensure that the traffic rate stays within the negotiated bounds. However, if the traffic is originated from a trusted source and admission controlled as defined in its SLS, no policing may be required. To guarantee the timely forwarding of the packets, a priority queue should be utilised [BakeF].

4.5.2 Multimedia Conferencing

The Multimedia Conferencing service class is ideal for applications that are real-time and have variable-length elastic traffic sources. The typical applications that fall in this category are those that are able to regulate their sending rate if the network is congested. Commonly, this refers to changing the sending rate of transmitter by altering the encoding parameters as a response to congestion notification received by the network (e.g., RTCP messages). In addition, this service class imposes access control and limitations on the volume of traffic that can be admitted into the
network. Within the DiffServ domain, AF PHBs and their traffic handling mechanisms (traffic conditioning, AQM and scheduling) should be used since they can provide differentiated bandwidth assurance for different priorities of traffic (i.e. AF41, AF42 and AF43). In summary, the service offered to the traffic of multimedia conferencing class is enhanced best-effort with controlled rate and delay. Both telephony and multimedia conferencing service classes’ applications fall into the interactive user traffic types.

4.5.3 Multimedia Streaming

Both Multimedia Streaming service class and Low Latency Data are types of responsive traffic. The Multimedia Streaming service class corresponds to applications that require near-real time packet forwarding of variable rate traffic sources that are less delay sensitive to delay than the interactive traffic class applications. In this class, traffic is usually buffered at the source/destination and therefore, is less sensitive to delay and jitter. AF with minimum bandwidth assurance is suitable for this service class. Characteristically, the traffic of this service class has variable sized packets, is bursty, but can tolerate around 2% packet-loss (depending on the application). The traffic handling mechanism of AF behaviour also applies to Multimedia Streaming service class applications but exceptional cases are possible (i.e., video surveillance and security streams). At the ingress of a DiffServ domain, their traffic is policed using single rate policers with a burst size control as specified in their SLSs. Like the case in the multimedia conferencing class, service offered to Multimedia Streaming traffic is enhanced best effort service with controlled rate and delay. However, the traffic source is not expected to respond to packet loss. Certain streaming applications (e.g. video surveillance application) are more sensitive to packet loss and delay; hence depending on their DSCP marking, they should be categorised to receive higher assurance of delivery (higher priority AF PHB). The AQM parameters should carefully be adjusted to account for the application sensitivity to packet losses and it can be replaced with a tail-drop mechanism.

4.5.4 Low Latency Data

The Low Latency Data service class refers to the treatment of elastic and responsive traffic that are typically from client/server based applications. Such applications usually have variable packet sized flows and are transmitted over TCP, which means that the source is capable of reducing its transmission rate when it detects or signalled packet loss. They require a relatively fast response with minimum bandwidth assurance. These requirements can be realised with AF service and appropriate traffic conditioning mechanisms. For instance, traffic flows treated under this class may be subjected to a single rate three-colour conditioner followed by a 3-level RED mechanism.
and a medium priority weighted queue. If peer-to-peer signalling flows are to be treated under this service class, then no AQM mechanism should be employed.

4.5.5 High Throughput Data

The High Throughput Data service class is ideal for responsive and loss tolerant applications that require timely packet forwarding of variable rate traffic sources. It is configured to provide good throughput for TCP longer lived flows. This service class needs to use the AF service with minimum bandwidth assurance for its flows. These flows are composed of variable length packets with bursts of TCP window size. The DiffServ domain conditioning requirements are same as Low Latency Data service class. It can be assumed that this class will consume any available bandwidth due to its high throughput, and packets traversing congested links may experience higher queueing delays and/or packet loss.

4.5.6 Standard Service Class

The Standard Service class refers to the Internet’s best-effort forwarding mechanism. The traffic streams that require Standard Service class treatment have no bandwidth, delay, loss, and jitter requirements. A certain percentage of forwarding resources should be reserved for this kind of traffic through a rate queueing mechanism. There is no requirement for this class of traffic to be conditioned in the DiffServ domain as excess traffic is dropped from its queue automatically. However, it is a good practice to employ a single rate RED algorithm to manage the queue depth and reduce the chance of congestion for the best-effort traffic.

4.5.7 Low Priority Data

The Low Priority Data service class is suitable for responsive applications that require no service guarantees. The LE PHB is suitable for this service class traffic. In the DiffServ domain, there is no requirement that conditioning of packet flows be performed for this service class. The fundamental service offered to this service class is best-effort service with zero bandwidth assurance. By placing it into a separate queue or class, it may be treated in a way that is consistent with a specific SLS.

4.5.8 Summary

In this section, seven service classes and the flow characteristics of their traffics have been introduced. Among these, Telephony, Multimedia Conferencing and Multimedia Streaming service classes are the focus of this research being presented here. Traffic from other service classes can be seen as the background traffic. From the information given above, it can be
understood that there is no generic way of configuring a DiffServ network to account for the requirements of all possible applications. Instead, the applications in demand can be identified and categorised into different service classes. A specific service class should be specified such that it considers the QoS requirements of the user, application and the network when configuring the traffic conditioning and scheduling parameters. In [BakeF], detailed examples of such scenarios were given and tables summarising the QoS mechanism used for each class were illustrated. Table 4.3 shows the summary of all service classes mentioned in Section 4.5 and the QoS mechanism that can used to realise such services, together with recommended DSCP values for every service class.

Nevertheless, it is difficult to argue that the table shown above is the generic way for service providers to classify the services that can be provided for the needs of different user applications. The design of differentiated services for traffic flows must reflect the objectives of the service provider or the network operator. In this respect, they must decide on the differentiation and granularity of aggregation levels that needs to be maintained for achieving these objectives. The detailed reason for this problem is explained in Section 4.7.

<table>
<thead>
<tr>
<th>Service Class</th>
<th>DSCP</th>
<th>Conditioning</th>
<th>Queueing</th>
<th>AQM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telephony</td>
<td>EF, CS5</td>
<td>sr+bs</td>
<td>Priority</td>
<td>No</td>
</tr>
<tr>
<td>Multimedia Conferencing</td>
<td>AF(<em>{41}), AF(</em>{42}), AF(_{43})</td>
<td>trTCM</td>
<td>Rate</td>
<td>Yes per DSCP</td>
</tr>
<tr>
<td>Multimedia Streaming</td>
<td>AF(<em>{31}), AF(</em>{32}), AF(_{33})</td>
<td>sr+bs</td>
<td>Rate</td>
<td>Yes per DSCP</td>
</tr>
<tr>
<td>Low Latency Data</td>
<td>AF(<em>{21}), AF(</em>{22}), AF(_{23})</td>
<td>srTCM</td>
<td>Rate</td>
<td>Yes per DSCP</td>
</tr>
<tr>
<td></td>
<td>CS4</td>
<td>sr+bs</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>High Throughput Data</td>
<td>AF(<em>{11}), AF(</em>{12}), AF(_{13})</td>
<td>srTCM</td>
<td>Rate</td>
<td>Yes per DSCP</td>
</tr>
<tr>
<td></td>
<td>CS2</td>
<td>sr+bs</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>Standard Data</td>
<td>DF</td>
<td>N/A</td>
<td>Rate</td>
<td>Yes</td>
</tr>
<tr>
<td>Low Priority Data</td>
<td>CS1</td>
<td>N/A</td>
<td>Rate</td>
<td>Yes</td>
</tr>
</tbody>
</table>

sr+bs represents a TB policing mechanism that provides single rate with burst size control.

Table 4.3: Summary of the Service Classes and the QoS Mechanisms Used [BakeF]
Chapter 4. Differentiated Services

The other important issue to consider while designing the differentiated services is the type of the access network. That is to say, if the access network is a wireless domain (i.e., UMTS), which identifies separate QoS parameters for the applications using this network, then a QoS mapping mechanism is required. This issue is explained in the following section.

4.6 Differentiated Services QoS Support in UMTS

Packet switched mode of the UMTS has adopted IP as the network transport level mechanism and promises guaranteed quality of IP multimedia services in both access and core networks. It provides support for different levels of QoS as required by the end-users and their applications. To provide this support, it has to employ a mechanism that can effectively differentiate between four UMTS QoS classes. The characteristics of these classes and the layered QoS architecture standardised by 3GPP have been introduced in Chapter 2. In addition, in its final phase of evolution, UMTS will become an all-IP network, which would need to make effective use of IP based QoS mechanisms in its domain. The DiffServ approach, with its scalable nature and differentiation ability, is well suited for networks with well-defined traffic profiles. Consequently, 3GPP has acknowledged that DiffServ can be used for providing transport level QoS guarantees in the IP based sections of the UMTS network (Universal Terrestrial Radio Access Network (UTRAN) and Core Network (CN)) [TS23.107], [TS23.207].

The UMTS QoS model is independent of the transport technology used in its core and radio access networks. UMTS is able to support end-to-end QoS guarantees for its packet domain, through the interaction of bearer services established between its modules at different layers. The UMTS bearer services support not only the specification of priority and bandwidth requirement type of parameters, but provide for a rich set of other service quality attributes. The QoS requirements of each traffic class are defined in terms of UMTS bearer service attributes. These can be categorised into radio access and core network bearer service attributes. The attributes are defined in a way that the system is not tied to any particular transport architecture, hence allowing flexibility for the network to evolve as the transport technology advances. The list of these attributes, their values and applicability to each traffic class is given in [TS23.107]. Therefore, the network operator's responsibility is to employ an appropriate mapping mechanism between these attributes and the chosen transport technology's QoS mechanism. However, the end-to-end service may be conveyed over multiple IP networks administered by different operators. Therefore, appropriate internetworking between UMTS core and other IP networks (e.g. the Internet) is also essential. Interoperability between operators is policy based and makes use of SLAs.
Chapter 4. Differentiated Services

The UMTS network can be seen as two separate DiffServ capable IP networks, one of them being the UTRAN and the other being the CN as depicted in Figure 4.13. DiffServ can be used in either radio access network or core network or both. In any case, the UMTS QoS classes need to be translated into appropriate traffic conditioning and PHB mechanisms. The mapping procedure from UMTS QoS classes to DSCP values is defined and controlled by the operator. Thus, the implementation of DiffServ functionality may exhibit differences amongst different operators.

In UTRAN, the edges of the network are defined by Node-Bs and RNCs. Therefore, the DiffServ border router functionality is realised within these components. Edge nodes act as the DiffServ boundary for the user traffic that is generated by UEs and perform the required traffic conditioning functions. The interior nodes interconnect Node-Bs, which could be spread over a large geographic area and possibly far away from the backbone. These nodes apply DiffServ PHBs on the traffic flows based on the mapping from the Radio Access bearer services to the DiffServ QoS classes. The network operator is responsible for its network’s resource and capacity management, hence appropriate traffic conditioning and PHBs should be designed by considering the planned traffic load of each UMTS traffic class.

In UMTS, QoS requirements are realised through the packet data protocol (PDP) context activation procedure. The UE sends a PDP context message to SGSN, which contains the desired QoS profile (conversational, streaming etc.) and relevant parameters (traffic handling priority, maximum bit-rate etc.). The QoS profile parameters of the PDP contexts are then mapped to IP transport parameters (DiffServ specific actions and routing control) through the use of Radio Access Bearer service [RaisV].
Chapter 4. Differentiated Services

If communication with an external IP network is required, GGSN provides the necessary interworking between the PDP context and the external network’s QoS mechanism. In other words, GGSN is the gateway of the UMTS to the external networks.

Between the UTRAN and the CN another QoS mapping operation needs to take place [TS23.107]. When a traffic stream passes from/to UTRAN to/from CN, the QoS characteristics corresponding to this stream should be translated to the new QoS system since both networks are defined by separate set of bearer services. In CN, if IP based transport technology is to be used, it is likely that the network operator can prefer to implement a different DiffServ network to support the QoS requirements of the traversing traffic flows. In this case, the traffic streams should be processed by the new DiffServ elements at the boundaries of this network (i.e., SGSN and GGSN). The traffic classes should be aggregated into a manageable set of net groups based on the translated QoS requirements. The aggregated traffic flows are then mapped onto appropriate PHBs on one-to-one basis. In [AghaF], a simple mapping mechanism between UMTS QoS classes and standard PHBs was introduced. It was reported that, since the UMTS network architecture and the user traffic profiles differ in specific ways than the Internet, common scheduling algorithms, such as WFQ or DWRR, may not be suitable to use. For a DiffServ based UMTS CN, the appropriate scheduling should be a fair, efficient and a simple algorithm that can provide link sharing and delay bounds independently. By this way, a diverse range of delay and resource allocations to various PHBs can be provided. Therefore, a DDB-FFQ (decoupled delay-bandwidth frame based fair queueing) scheduler should be used [AghaF2].

In UMTS, guaranteeing end-to-end QoS means that the network operator needs to support interoperability with external IP (e.g., the Internet) networks. This requires another mapping mechanism to be in place so that the relevant CN QoS classes and service attributes can be translated into external network’s QoS definitions and mechanisms. This is a rather challenging issue due to some topological differences between the UMTS and the Internet. In UMTS, for example, there are four QoS classes each of which has its own distinct requirements from the network. The wired domain of the UMTS network is well provisioned and is assumed to introduce minimal delay or loss under normal operation circumstances. For this reason, the DiffServ functionality can be utilised to achieve specific delay, jitter and loss bounds for users’ applications. The users are charged based on the amount of data they send/receive or the duration of the call they make. In other words, the access to the network resources is limited with a price constraint mechanism. In the Internet, on the other hand, applications cannot be generically categorised into a specific number of service classes since there are numerous applications with varying characteristics and many users with heterogeneous capabilities (e.g., access bandwidth, terminal type etc.). In addition, due to large number of applications, the traffic streams tend to be
more coarsely aggregated into different PHBs. Lastly, typical charging strategies are based on fixed charging schemes which do not impose any well defined restriction on the traffic volumes introduced into the network. Therefore, although IP can ensure interoperability at the network layer, specific mechanisms must be in place to perform the translation of UMTS QoS classes and service attributes to external network's service classes and traffic engineering parameters. Trying to address some of these problems, [ManiS] proposed a DiffServ architecture, which maps the UMTS DiffServ traffic PHBs onto an external IP network's corresponding DiffServ PHBs. The mapping mechanism takes into account the characteristics of UMTS traffic classes and determines how the relevant bearer service attributes are translated into the service parameters (e.g., traffic conditioning) of the external network in order to achieve end-to-end QoS guarantees.

4.7 Differentiated Services for Multimedia

As explained in Chapter 2, continuous media applications have varying requirements from the transport network. As a result, DiffServ model is useful for servicing different kinds of multimedia applications with diverse characteristics. If EF service is used, continuous media applications can be provided with tight delay/loss bounds; hence, the issue of concern is how to configure the traffic conditioning parameters to abide by the promised service guarantees stated in the SLA. On the other hand, the main challenge in a DiffServ network is how to make use of the AF for service differentiation. Continuous multimedia applications, which are less time critical are somewhat resilient to packet loss and delay violations, and can be forwarded with AF service if the priorities are carefully configured (i.e., as shown in Section 4.5).

The fluctuations in the network resources may cause increased delay, jitter and packet-loss experienced by the applications. If BE forwarding is used, the delay, jitter and loss experienced by the applications become difficult to control. In this case, the perceptual quality might be degraded due to the congestion caused by applications competing for the same bandwidth. Therefore, it is essential to provide some means of prioritised treatment for multimedia applications to minimise these adverse effects. So far in this chapter, the mechanisms that can be utilised to construct a system of differentiated services for various traffic categories have been presented. In this section, the appropriateness of the general trend of classifying multimedia applications into specific service classes is discussed. Contrary to this, it is argued that the extracted information from the multimedia applications can be used to bias the traffic conditioning and the forwarding attributes of the DiffServ network. This is performed by making use of a model, which analyses the multimedia content during the encoding process and enables the perceptually more critical information to be given higher priority in transmission. However, this requires a finer granularity
of aggregation to be in place. The consequences and trade-offs of such approach are discussed below. The intra-flow packet differentiation can yield improved performance for all types of multimedia applications, yet the examples given below are limited to the video-based applications.

### 4.7.1 Granularity of Aggregation

The network operator needs to map various traffic types on different behaviour aggregates, which are expected to receive a predefined traffic conditioning and forwarding behaviour based on their DS markings. For this purpose, as introduced in Section 4.5, an example model for defining the traffic categories and service classes can be taken as a reference. In this approach, similar applications are categorised into the same service class under the assumption that their service requirements and their sensitivity to changing network conditions are similar. The systematic allocation of network resources amongst various service classes promises better performance than Internet’s default forwarding behaviour (BE). However, even such allocation and service differentiation may be regarded deficient in addressing the requirements of complex multimedia applications. That is to say, this model is constructed by considering the timeliness and packet-loss requirements of the applications that are assumed to have the similar traffic characteristics as defined in [G1010] and [Y1541]. An aggregated traffic stream is subject to a specific packet treatment policy applied at every hop. This treatment offers limited differentiation amongst the packets of the same aggregate (e.g., a multi-stage AQM mechanism applied on the AF’s drop-precedence). This means that a coarse granularity of aggregation would result in an indiscriminative packet treatment amongst different multimedia streams being aggregated together, as well as amongst the packets of the same multimedia codec (no unequal protection support). However, if the granularity is made finer, the differentiation between different streams would be increased, which could enable the service provider to devise more specialised services for multimedia streams. For this reason, it can be said that the granularity of aggregation is a significant factor in the perceived quality of the application. Therefore, if the perceived service quality is the main concern of the network operator, a finer granularity of aggregation should be provided.

Nevertheless, enhancing the granularity would mean that more number of PHBs should be defined and applications should be made aware of the network capabilities as well as networks made aware of the traffic contents they carry. The disadvantage of such approach is that increasing the number of forwarding behaviours in the network and additional intelligence required, would bring in added management complexity and may degrade the end-to-end delay performance. Moreover, considering the high volumes of traffic being handled at the core of the
DiffServ network, the number of PHBs should be kept small in order to match the packet processing speed with the transmission rates. The other problem that is likely to be experienced would be the difficulty of mapping of the QoS attributes of each PHB between different network domains. Standardisation of PHBs helps the service providers to define appropriate mapping rules that minimise the QoS degradation when their traffic requires traversing between different administrative domains. However, a fine granularity of aggregation may require specialised PHBs that are not standardised and difficult to interpret by the operators of other DiffServ domains.

Clearly, there exists a trade-off between less network complexity with a coarse aggregation and a better perceptual performance with a finer one. Exploiting the properties of the multimedia information for introducing sophisticated traffic treatment mechanisms presents an open area for research. The research activities about how to make use of the DiffServ capabilities to provide QoS for multimedia applications can be categorised under two schools of thought. These are:

- Prioritisation based on packet contents
- Prioritisation based on traffic category

### 4.7.2 Prioritisation Based on Packet Contents

The traffic patterns produced by multimedia codecs vary greatly with the type of the information (e.g., audio, speech or video) being processed. This means that given the same network conditions, different application streams aggregated into the same PHB will be affected differently due to the contrast in their resilience to packet loss, corruption, jitter, and delay. Moreover, different packets belonging to the same application may carry different importance with regards to their effect on the perceived quality at the end-user. For instance, a motion compensation based video coder (e.g., MPEG-4) can produce a traffic stream whose packets can be assigned different priorities based on:

- Video object layer (VOL) level: e.g., base layer or enhancement layer.
- Frame level: e.g., I, P or B frame.
- Video object level: e.g., foreground or background.
- Group of Blocks (GOB) level: e.g., which GOB in a frame has higher distortion effect.
- Video packet (VP) level.
- Macroblock level: e.g., I-block, P-block or motion vector scale.

 Although this list can further be expanded, the main issue being addressed here is how to utilise the network's service differentiation capability by making use of the characteristics of the multimedia applications. For instance, some of the information available at the encoding stage can be used to develop effective packet marking strategies. One possible way of doing that would be...
to incorporate a distortion analysis model in the encoder. Such model would analyse the effect of packet-loss on the perceptual quality and mark the packets accordingly. While it is possible to develop a low complexity algorithm that would not hinder the real-time presentation of applications (e.g., video conferencing), more complex mechanisms can be used which can analyse the video content offline and differentiate priorities on demand (e.g., video streaming).

In [VitoF], a perceptual quality based adaptive packet marking mechanism was introduced. In this work, the packet marking process was provided with feedback from the Group of Pictures (GOP)/frame level distortion computation algorithm. This algorithm is able to analyse the packet loss distortion effect on each frame, as well as the distortion in the future frames due possible error propagation. Based on the analysis, a fixed percentage of those video packets with relatively more important information were marked with high priority. High priority packets were transmitted over the premium service that can offer a guaranteed QoS (i.e., using EF PHB).

Video Object (VO) based differentiation is also another promising approach. [HouY] has demonstrated the advantages of VO based video packet prioritisation. In this work, the foreground object was segmented from its background and assigned with a higher priority for transmission (i.e., low loss probability). The foreground and background VOs are shown in Figure 4.14. The service differentiation between different classes was achieved through the use of a modified version of a RED algorithm and the AF mechanism.

![Fig 4.14: (a) Original Akiyo Sequence, (b) Background VO, (c) Foreground VO [HouY].](image)

It was shown in [HouY] that in the cases of network congestion, the perceptual quality of the video stream could be maintained at an acceptable level if service differentiation is used. Figure 4.15 demonstrates the benefit of VO prioritisation. A similar but more detailed approached was also presented by [ShaoH]. A more elaborate packet marking strategy, which takes magnitude and direction of motion vectors, macroblock encoding types, and delay/loss requirements into account was presented by [ShinJ]. Based on these parameters, a relative priority index is assigned to each
H.263 video packet, which designates the required services from the network. The service differentiation is provided by a variant of the WFQ algorithm coupled with the RIO active queue management mechanism. Experiments showed that considerable perceptual quality improvement is possible by using this packet level differentiation strategy.

![Images of video frames](a) (b) (c)

Fig 4.15: (a) 20kbps Encoded with no Packet loss, (b) Both Foreground and Background Subject to Average 7.4% Packet Loss, (c) DiffServ Case with no Packet Loss for the Foreground and Average 9.5% Packet Loss for the Background [HouY].

### 4.7.3 Prioritisation Based on Traffic Categories

The more generic way of providing differentiated services is based on traffic categorisation as exemplified in Section 4.5. However, the model given in Section 4.5 is designed to serve all possible types of applications that are prominent in today’s networks. In practice, service providers are not required to provide all eight types of service classes. Thus, they may choose to introduce only a few levels of differentiation that can adequately serve the needs of their customers. The end-user applications can be grouped into a few number of service levels based on the assumption that they have similar requirements from the network and they have the similar characteristics (e.g., delay, jitter, packet size, error resilience etc.). When deciding on the type and number of differentiated service levels required in the network, the service provider may make use of traffic measurement tools that can provide extensive information about the type, profile, resource utilisation and traffic volumes of users’ applications.

If, for example, real-time and low-latency multimedia applications need to be differentiated, three different service classes can be defined [TsolE]. The first service class is suitable for applications (e.g., VoIP and interactive multimedia) that require a circuit-switched network like performance (low delay, very low jitter and packet loss) from the network. The traffic flows benefiting from this service are expected to be near-constant bit-rate with small packet sizes and bandwidth requirements. Therefore, this service class defined in terms of a short-length priority queue and
traffic conditioning rules that provide maximum end-to-end delay as 150 msec and packet loss rate of $10^{-6}$. The second service class makes use AF mechanism and is appropriate for low-latency multimedia application (e.g., multimedia streaming). The traffic flows associated with this service class can be variable bit-rate with larger packet sizes. The targeted delay and packet loss figures for this service class should be 250 msec and less than $10^{-4}$ respectively. The third service class is dedicated for responsive flows, which are served on best-effort basis. The implementation details of these service classes and the others that are not mentioned here are given in [D1301]. As presented in [TsoIE], these services classes, when coupled with appropriate admission control, traffic conditioning and scheduling mechanisms, can efficiently provide improved QoS for multimedia services as compared to IP networks’ default forwarding behaviour.

4.7.4 Discussion

Granularity of aggregation is a significant factor in designing a QoS enabled system. It defines the service differentiation scale of multimedia information with respect to other types of traffic. While aggregating applications with similar properties is the essence of DiffServ, the granularity of aggregation is what determines the quality provided in resource-limited networks. Making use of packet level differentiation within the same multimedia stream improves the given service quality but requires the network to keep an increased number of state information at the borders, and a larger set of PHBs at the core. In an environment (i.e., the Internet) where there are numerous different application developers, it would be very costly for the service providers to manage their network where every multimedia codec has its own defined set of differentiated services. Therefore, the challenges here can be defined as:

- How to find the optimum granularity of aggregation.
- How to establish the right balance between packet level categorisation and traffic aggregation.

In the view of the research work presented in this thesis, the traffic aggregation does not necessarily need to consider packet-content level prioritisation. Instead, the applications should be classified based on their basic networking and presentational characteristics (as explained in Chapter 2). However, service providers can also choose to map different applications into the same PHB based on the promised performance guarantees offered to their customers. In this sense, there is not any granularity of aggregation that is favourable to others under any circumstances. Nevertheless, it is claimed in this thesis that the DiffServ networks should keep the number of available PHB limited in order to preserve network core simplicity. Application specific service differentiation, however, can be introduced through the use of active networking
technologies at the edges of the networks. A demonstration of executing such services for a video application in a DiffServ aggregate is presented in Chapter 5.

In the next section of this chapter, a performance analysis of introducing application level service differentiation for multimedia is presented. Later in Chapter 5, this scenario is expanded to include the effects of introducing active services at the network edges.

4.8 A Performance Analysis of Differentiated Services for Multimedia

In this thesis, the aim is not to find solutions to a particular network configuration problem but to demonstrate the possible advantages of differentiated services on the quality of the multimedia applications. For this purpose, a practical implementation of DiffServ disciplines was required. Such practical implementation can make use of a simplified model that is capable of demonstrating the usefulness of differentiating the services given to different traffic aggregates. To achieve that, a networking scenario with a limited number of users and network nodes is sufficient. The study presented in this section focuses on multimedia transmission over a congested IP-network. The advantages of using DiffServ traffic control mechanisms are highlighted by comparing the performance of the DiffServ model with the best-effort model.

4.8.1 The Simulation Model

This section describes the network topology chosen for performing the different simulation scenarios. The OPNET simulation tool was used for building the simulation model. The experimental network topology is depicted in Figure 4.15.

In this model a VoIP source (VoIP_Src), a Video Conferencing Source (Video_Conferencing_Src), a Video Streaming Source (Video_Streaming_Src), and a Background Traffic Source (BG_Traffic_Src) was used to create the network traffic. Each source has a corresponding destination at the other end of the network. There are four routers in the network; two of them representing the edge routers of the DiffServ domain (Edge_Router1 and Edge_Router2), and the other two representing the Core Routers (Core_Router1 and Core_Router1). Each traffic source and the destination are connected to their edge routers with 2Mbits/s links. The capacity of the each link represents the bandwidth made available to the applications that fall into different service classes (i.e., telephony, multimedia conferencing, multimedia streaming, and standard service class). On the other hand, edge routers are connected to the network core with 10Mbits/s links. A 2Mbits/s link between the core routers was chosen to create the simulation bottleneck.
Chapter 4. Differentiated Services

The VoIP source was configured to model the traffic pattern generated by the iLBC codec at 15.2kbits/s with 304bits/packet [AndeS]. The video conferencing source was designed to produce QCIF (Quarter Common Intermediate Format) resolution and 10frames/s MPEG-4 traffic at a constant bit-rate of 128kbits/s. The GOP structure for the video stream was chosen as (19, 1), representing the use of an I-frame at every 2 seconds. The video frames are packetised into 576bytes packets. The model used for video streaming application is similar to the one used for the video conferencing application. However, the video streaming source modelled a variable rate MPEG-4 stream encoded at average 450kbits/s. Both of the video sources represented the “Foreman” video sequence. The same sequence for both of the video applications was used in order to be able to perform the comparison of the performances of different video applications with each other under the same network conditions.

The background traffic represents the traffic created by all other applications (i.e., mixture of other UDP and TCP traffic) that are neither time-critical nor have paid for the privileged service. The background traffic was emitted into the network at average bit-rate of 1.31Mbits/s. The modelling of the background traffic is based on the characteristics of the Internet background traffic models. The methodology used for creating the background traffic is explained in the following subsection.

All the routers, except Core_Router1, can forward the incoming traffic to an output port at 10Mbits/s and have a buffer size of 100kbits. For the purpose of simulating the network...
bottleneck, the link connecting the two core routers and the servicing capacity of the Core_router1 is limited to 2Mbits/s.

### 4.8.1.1 Modelling the Background Traffic

In general terms, the background traffic corresponds to aggregated IP traffic existing in the network backbones. In the simulation model described above, it represents the cumulative traffic created by a number of users running various different applications. Such aggregated traffic patterns tend to show long term correlations between the packet inter-arrival times and are extremely bursty in nature. Based on the studies performed in this field, the background Internet traffic is generally characterised by heavy tails [LelaW], [PaxsV], [ThomK]. The Pareto distribution with a shape parameter $k<2$ is a heavy tailed distribution which has an infinite variance. Therefore, it can be used for modelling the background traffic [BourC], [JamiS].

In light of these facts, Pareto distribution was used to create the background traffic. Thus, if $X$ is a random variable then the Pareto distribution is defined with the following probability distribution function:

$$P(X > x) = \left( \frac{x}{x_m} \right)^{-k}$$

(4.6)

where $x$ is any number greater than $x_m$, which is the (necessarily positive) minimum possible value of $X$, and $k$ is a positive parameter. The Pareto distribution is parameterised by two quantities, $x_m$ (location parameter) and $k$ (shape parameter). The probability density function is then:

$$p(x | k, x_m) = k \cdot \frac{x_m^k}{x^{k+1}} \text{ for } x \geq x_m$$

(4.7)

The expected value of a random variable using Pareto distribution stands for the mean inter-arrival time between consecutive packets and is calculated using the following formula:

$$\mu = \frac{x_m k}{k - 1} \text{ for } k > 1$$

(4.8)

and the variance is defined as:

$$\sigma^2 = \frac{x_m^2 k}{(k-1)(k-2)} \text{ for } k > 2$$

(4.9)

In the OPNET environment, the background traffic source module is modelled by using 100 separate Pareto traffic sources. Each of these sources is configured to produce packet traffic based
on their input \(x_m\) and \(k\) parameters. The synthesis of the packet sizes is implemented in such a way that it resembles the mixed Internet traffic as defined in [ThomK]. In that respect, 50% of the background traffic is composed of 40-byte packets, 40% is 576-byte packets, and 10% is 1500-byte packets.

### 4.8.2 Experimental Procedure

Two sets of experiments were devised to analyse the performance of the DiffServ with respect to the Best-Effort forwarding service. The first experiment involved the simulation of the Best-Effort service with the aforementioned traffic patterns. For this purpose, a simple FIFO scheduling scheme with a single buffer was implemented in each router. The servicing rate of the routers (except for the Core_Router1) was set to 10Mbits/s. The experiment was run six times with the background traffic volume increasing by 3-4% at each run. At the onset, the total average traffic volume streamed into the network was 1.90Mbits/s. Later, with the increment in the number of sources used in the background traffic model, this figure was progressively raised to 2.04Mbits/s, 2.09Mbits/s, 2.17Mbits/s, 2.22Mbits/s, and 2.28Mbits/s. The resulting packet-losses, end-to-end delay figures, and the consequent degradation in the received multimedia quality were recorded.

The second experiment was aimed at demonstrating the effects of utilising the DiffServ functionalities in the network. For this purpose an AF based traffic classification was made. Each traffic category was marked with a specific label which indicated their corresponding PHB. Based on these classifications, each traffic source was assigned with a specific share of the total bandwidth. It was assumed that the volume of traffic emitted by each traffic class is in conformance with their TCAs. The sharing of the bandwidth is achieved by using a DWRR scheduler at the network routers. The total buffer size used by the scheduler was same as the single FIFO scheduler case (i.e., 100kbits). The scheduler made use of 4 finite length FIFO queues, whose buffer sizes were directly proportional with the traffic volume of their corresponding traffic. The bandwidth (in terms of queue weights) and the buffer space reserved to each traffic flow are illustrated in Figure 4.17.
As can be seen from this figure, the scheduler is configured to allocate the available bandwidth between the different forwarding services. Since the volumes of the multimedia traffic are within their pre-specified (i.e., TCS) levels, no policing or shaping is required. However, the Best-Effort traffic is subject to tail-drop mechanism when it demands more than its share of available bandwidth.

The simulation duration for both experiments was limited to 14 seconds due to the length of the "Foreman" test sequence used to assess the effect of packet loss on the video applications’ visual quality. The multimedia sources started to emit packets into the network from the beginning of the simulation until the 13.4th second. On the other hand, the background traffic was configured to start at the 2nd second of the simulation time and at the end of the simulation.

4.8.3 Results and Discussion

In the first experiment, all of the network routers were equipped with a single FIFO queueing mechanism. As the background traffic was gradually increased, Core_Router1 started to drop packets due to its limited queue size and service rate. Since the background traffic generation was based on a random process, each experiment was run for 15 times with different seeds in order to obtain more reliable results. The collected results were averaged over five simulations. The average packet loss observed for each multimedia application at different background traffic loads is given in Figure 4.18.
As can be seen for Figure 4.18, the VoIP application has lost the least number of packets amongst other multimedia applications. This is because, the VoIP application has smaller packet sizes (i.e., 40 bytes) and its source rate is significantly lower than the other two multimedia applications. The video applications, on the other hand, were packetised using the same packet size (i.e., 576 bytes). Therefore, the difference in packet loss is related to the difference in the packet inter-arrival times. Since the VBR VS application had a higher source-rate, the average inter-arrival time between its packets were lower than that of VC. Although the number of packets lost for the VC application was relatively less than that of VS application, its packet loss rate was measured to be higher. Since the packet sizes were the same for both video applications, the difference in packet loss rates is related with the packet inter-arrival times. Considering the bursty nature of the BG traffic, the congestion period is composed of the transient but frequent Core_Router1 buffer overflows. Hence in this case, the probability of the application with faster date to lose packets is smaller. In the experiments performed in this section, the congestion collapse situation was not examined. If congestion collapse had occurred, all the applications become equally affected by the congestion.

The end-to-end delay for each application was also measured. This was performed by measuring the delay experienced by each packet arriving to its destination, and averaging the obtained values over the total number of received packets. It can be observed form Figure 4.19 that the end-to-end delay increases as the congestion level increases. However, from the point where the background traffic rate is around 1.486 Mbits/s onwards, the increase in the delay comes to saturation since the router queue operates close to its maximum capacity. The end-to-end delay experienced by the packets of the Video Conferencing application is higher than other two multimedia applications.
since its average inter-arrival time between the packets are higher. Thus, VC packets are likely to wait longer in the queue since the probability of finding a rather less occupied queue is low.

Figure 4.19: Average End-to-End Delay Experienced by Multimedia Traffic with the Increasing Volume of Background Traffic

PSNR measurements were performed to measure the effects of packet losses on the video quality. The results of these measurements are depicted in Figure 4.20. This figure was drawn by selecting an outcome of one of the 15 runs made for each background traffic rate. The selected packet loss ratios were the ones that are closest to the average packet loss rate. Following this, the selected packet loss rate and patterns that were recorded at the destination points were applied on the encoded video streams. The corrupted video streams were then decoded and the PSNR measurements were applied. The result of these experiments revealed interesting findings. As can be seen from the Figure 4.20, after the background traffic level reaches to 1.44Mbits/s, the degradation in the quality of the Video streaming application is worse than that of Video Conferencing, despite suffering from less percentage of packet loss. This shows that the application characteristics are an important factor in the perceived quality of the multimedia applications. Another conclusion that can be drawn from this result is that increasing the application bit rate does not necessarily increase its robustness against packet losses.

To measure the perceived quality of the VoIP application under different levels of packet loss, Mean Opinion Score (MOS) tests were required to be performed. However, a comprehensive study of the performance of the iLBC codec under different rates of packet loss was given in [ILBC]. Therefore, the performance of VoIP application is evaluated with reference to the results
published in [ILBC]. Based on these figures, the performance evaluation of this application is depicted in Figure 4.21.

![Figure 4.20: Quality Degradation in the Video Traffic with the Increasing Volume of Background Traffic](image)

The MOS results shown in Figure 4.21 reveal that the VoIP application is able to preserve its intelligibility even when the VC and VS applications become unusable. One reason for that is because the iLBC codec is specifically designed for low bit-rate and real-time voice communications over the Internet. While the VoIP applications usually have a basic format, video based applications can have significantly different compression levels, resolutions, frame rates, object contents etc. In addition, most of the video based applications are optimised for reasons other than robust transmission. Therefore, network QoS support for such applications is essential.

The same simulation scenario was repeated in the second experiment. Only this time, the DiffServ functionalities were implemented in the routers instead of the Best-Effort system which employed a single FIFO queue. By using the DWRR scheduling mechanisms, each flow was assigned with a certain portion of the total bottleneck bandwidth. Based on the measurements made before injecting the excess background traffic in the network, the scheduler was programmed to serve the background traffic for the 66%, Video Streaming 25%, Video Conferencing 8%, and VoIP 1% of the total scheduling time. As a result, no packet drop could be observed in for the multimedia applications throughout the simulations performed for the DiffServ case. In other word, increasing the background traffic had no effect on the multimedia applications in terms of packet loss. The only packet loss observable was for the background traffic.
However, the use of this scheduling mechanism incurs a delay overhead. That is to say, for the non-congested case (i.e., when the background traffic volume is around 1.31 Mbits/s), the average end-to-end delay has increased by 7ms for the Video Conferencing, 7ms for the Video Streaming, and 17ms for the VoIP applications. Nevertheless, unlike the case in Best-Effort approach, these figures remain somewhat constant even when the background traffic volume is increased. The delay performance of the multimedia applications is depicted in Figure 4.22.

So far in this section, the advantages of introducing a level of service differentiation in the network are discussed. However, the scenario presented here considered no competition between the individual streams contained within the same traffic aggregate (e.g., as in different video streaming applications competing for the same network resources). In fact, unless the network resources are over-provisioned, AF service cannot guarantee QoS for its aggregates but promises higher service reliability than default behaviour (i.e., best effort forwarding). When AF aggregates are highly utilising the available network resources, certain policing and shaping rules should be applied (as specified in their TCS). The basic implication of this would be dropping packets at the network boundaries. Packet drops have adverse effects on the perceived quality of continuous multimedia applications. Therefore, unless the multimedia is adaptable to varying network conditions using hierarchical coding schemes (e.g., as in layered video coding [RejaR], FGS MPEG-4 [HsiaH], multiple description coding [WangY2]), packet losses should be minimised.
Therefore in the next chapter, the advantages of employing transcoding services at the network congestion points are investigated.

![Figure 4.22: Delay Performance of the Applications Using the DWRR Scheduler](image)

**Figure 4.22:** Delay Performance of the Applications Using the DWRR Scheduler

### 4.9 Concluding Remarks and Recommendations

This chapter has introduced the DiffServ IP QoS architecture which enables service differentiation for different categories of end user applications. The motivation of the DiffServ comes from the need for providing better QoS to multimedia applications and creating a service level differentiation for the market place. The QoS given to user applications is defined by SLAs, which basically specifies the amount of resources allocated and the traffic conditioning rules applied for the user applications based on a specific charging scheme. Individual traffic streams that have similar QoS requirements are multiplexed into traffic aggregates. The conditions specified in an SLA is realised by applying particular forwarding behaviours for the traffic aggregates.

A DiffServ domain is characterised by a cloud model with edge routers at the borders and the core routers in the centre. Edge routers are responsible for performing complex classification and conditioning operations, while the core routers, due to the large volumes of traffic they carry, are only expected to forward the incoming traffic towards their destination. The forwarding procedure involves differential queueing and complex scheduling operations, which isolate the QoS received by a particular traffic aggregate from the others.
A number of different forwarding behaviour have been standardised for the DiffServ networks. Amongst these, EF and AF are the most commonly referred and studied ones. However, the investigations performed by the Internet2 consortium have revealed that EF based service provision has a very high cost relative to its perceived benefits. Therefore, the main interest among the research community has shifted towards exploiting the potential advantages of utilising an AF based service provision. In parallel to this view, the study reported in this thesis has mainly focused on the approaches employed for providing service differentiation to different multimedia applications by using AF based mechanisms.

One of the important challenges in servicing multimedia applications in a DiffServ network is determining the right granularity of aggregation. A coarse granularity of aggregation means that the packet treatment amongst different multimedia applications of the same aggregate will be indifferent. This approach does not allow the applications to make better exploitation of the network’s service differentiation capabilities. Therefore, the perceived QoS at the user end would differ amongst different applications. Nevertheless, coarse granularity means simplicity and the fact that such network configuration can still offer better than best-effort service is an attractive option for the service providers. On the other hand, if the granularity is made finer, the differentiation between different streams would be increased and even intra-stream differentiation (i.e., packet-content based) would be realisable. While this could optimise the QoS received from the available network resources, it would also bring additional management complexity and perhaps the need for standardising new PHBs. In conclusion, the granularity of aggregation should reflect the objectives of the service provider and can either be coarse or fine depending on the performance targets specified for user applications.

To demonstrate the possible advantages of introducing DiffServ in the networks, a series of experiments were performed. In first phase of the experiments involved the testing of the perceived quality of the multimedia applications (VoIP, VC, and VS) with best-effort service forwarding was performed. In the second phase, each application was aggregated into a separate service class based on the arguments made in Section 4.5.

In the best-effort scenario, it was observed that if applications are unable to regulate their transmission rates in the case of network congestion, they will be subjected to packet losses which have serious effects on the perceived application quality. In the experiments performed, the packet loss rates experienced by the applications showed some differences. These differences were associated with the packet sizes and the packet inter-arrival times. As the packet inter-arrival time increases, the probability of finding a space in a congested router buffer decreases. If the packet size is decreased, the probability of placing the packet into the buffer increases. Therefore,
compared to the VS application, VC suffered a higher rate of packet loss. This is due to the longer inter-arrival times between the packets of the VC application. If the VC had used smaller packet sizes, the packet loss rate would have decreased but this time more bandwidth would have been used. Another important observation was made about the levels of quality degradation for the video-based applications under the same channel conditions. It was seen that after a certain level of packet loss rate, the degradation in the quality for the VS application was worse than the VC, even though VS application was encoded at considerably higher rate. This shows that in the case of packet-losses, there is a limit to the improvement that can be achieved by increasing the application’s bit-rate. In other words, increasing the bit-rate above certain threshold results in quality degradations. Therefore, there is an optimal average bit-rate, which is somewhat independent of the packet loss rate but more dependent on the spatio-temporal characteristics of video sequence. As a result, it is clear that providing a means for congestion adaptation for the applications is essential for achieving QoS enabled content delivery.

In the second phase of the experiments reported in Section 4.8, the best-effort forwarding was replaced by a DWRR scheduling mechanism which was used to realise the basic DiffServ functionalities. By allocating fair portions of the available bandwidth to individual applications, it was observed that the packet loss events for the multimedia applications can be eliminated. In this case, the increase in the background traffic volume could only affect the packet loss experienced by the BG traffic. In case when the BG traffic was at its lowest level (no congestion case), a slight increase in the end-to-end delay performance of the multimedia applications was also observed. This is due to the Round Robin (RR) operation of the selected scheduling mechanism. In this scenario, no competition between the streams of the same aggregate was assumed. In the next chapter, this issue is further investigated and the advantages of introducing congestion control are introduced.
Chapter 5

5 Active Services QoS Support

5.1 Overview

As described in the previous chapter, providing differentiated treatment to multimedia traffic flows is an effective approach in providing QoS enabled transmission of the user applications. Nevertheless, the QoS treatment that can be given to multimedia packets in a DiffServ capable network node is mainly limited to a number of traffic conditioning, queuing and forwarding mechanisms. Furthermore, cross-network QoS support is rarely available in heterogeneous networks for various administrative and technical difficulties. Such difficulties can be alleviated with the use of active networking technologies which can extend the capabilities of IP QoS (e.g., DiffServ) architectures beyond their standardised operation.

In particular, multimedia applications can greatly benefit from various active network services deployed inside the network, which can enhance the QoS provided by the transport network. As well as being applicable to various QoS related issues (i.e., as explained in Section 3.6.1), active services can help to provide seamless operation between heterogeneous network domains.

This chapter presents active services and discusses the vision in which they provide QoS support for multimedia applications. In particular, those services which involve the transcoding operations are introduced, and their performance in terms of providing QoS support for video applications is demonstrated.

5.2 Active Services

Active services are software modules that are installed in active network nodes to execute a specific function. Unlike the case in DiffServ, active services are not only capable of providing QoS support per-flow basis, but can also provide customised processing at the packet level. In the context of the research work presented here, they are services that are capable of processing the multimedia streams inside the network nodes. Such services provide data manipulation algorithms
that can modify the contents of the packets in traffic flows, and operate based on the requirements
dictated by application, users, and service providers.

The scope of these services can range from the deployment of new communication protocols in
the network to manipulate the contents of individual packets. In this thesis, active services are
considered to be a set of tools that affect the QoS provided for the multimedia applications. On
this basis, they can be classified into several main classes of services.

- **Filtering Class Services**: Priority based filtering, selective packet dropping, dynamic
  packet classification and scheduling, media caching.
- **Routing Class Services**: Active routing, resource reservation, adaptive multicasting.
- **Combining Class Services**: Media synchronisation, media multiplexing.
- **Transcoding Class Services**: Rate-control, resolution manipulation, error-resilience,
  syntax translation, transcoding, and application layer multicast.

These services can be composed and deployed either by the user or the service provider. The
services managed by the service provider control the network resources shared amongst the users
and provide flexibility in managing the network. Users, on the other hand, can utilise certain
services to alter the way available/reserved active node resources are used, and obtain the required
customisation on the received multimedia information. In the view of the author, the
programming model used for the active services should follow the code reference approach for
the reasons explained in Section 3.3.3.

The active service classes can be categorised as being network-centric or user-centric depending
on which communication layer they are activated. That is to say, active services associated with
the filtering and routing class services are network-centric and usually deployed at the network
and transport layers. On the other hand, combining and transcoding class services are user-centric
and usually deployed at the application-layer. In the research work presented here, the
investigations were concentrated on the development of application-layer services and in
particular transcoding class services.

The user-centric services have the property of being able to directly affect the QoS perception of
the users. These services are usually associated with signal processing operations and can be
highly demanding in terms of processing power and memory requirements from the active node.
Packet networks can usually be represented with the cloud model where its core routers are
located in the centre and the edge routers are placed at the network boundaries. Therefore, user-
centric services are more likely to be performed at the network boundaries. This is due to the fact
that the boundary routers carry less volumes of traffic and are able to reserve the necessary computing resources for flows which require active processing. The core routers on the other hand may be preferred to be stateless and perform only forwarding of the traffic flows. However, in order to realise the full benefits of introducing active services in the networks and providing QoS support on end-to-end basis, the active services must ideally be realisable throughout the network at every necessary network node.

One of the major objectives of the research work presented in this thesis was to develop application-layer services which can take advantage of the processing capability in networks for improving the QoS perceived by the end-users. As a result, the focus of research was on the development of advanced services using the functionalities provided by the active platforms. Therefore, a fully active network was assumed to be at the author’s disposal and the author relies on the respective efforts in the development of a functional and well-performing active networking platform.

5.3 Transcoding Class Services

The distribution of real-time multimedia information to both fixed and wireless network clients requires the networks to provide services for content adaptation. This is due to the vast amount of network heterogeneity, diverse user terminal capabilities and possible service level agreements with the local service provider. Considering the increasing demand on video communications, comprehensive network services need to be designed for content adaptation. Such services can become the interface between the applications and users by compensating for the problems introduced by the transport media and the application-level incompatibilities between the users. However, content adaptation services have to account for the time-sensitive properties of the multimedia applications. That is to say, they should not induce any significant delay overhead, which may hinder the interactivity and/or the natural continuity of the multimedia presentation.

As mentioned earlier in Chapter 2 proxy/gateway approaches could be the answer for the content adaptation requirements in the networks. Nevertheless, these approaches suffer from scalability problems and may require new protocols that can ensure their efficient operation in the traditional networks. The user interaction with these components is limited, and services received by the users cannot be changed dynamically to adapt to changing network conditions and/or user requirements. In this sense, active networking can be considered as the ideal medium for content adaptation since it is capable of providing customisable AP inside the network. In fact, one of the main motivations behind developing the AN concept was to create a system where the ultimate
support for multimedia applications could be provided. This would be provided by enabling user controlled complex processing operations (e.g., transcoding) to take place in the network [TennD], [TennD2], [KulkA], [AmirE]. In video communications, transcoding is the main element of content adaptation. Realising this fact, a number of transcoding services were proposed in the active networking literature [DuysB], [GuoJ], [KhanJ [OttM].

In the view of the research presented in this thesis, video transcoding operations form the basis of the active services, which provide support for QoS in video communications. Therefore, the video transcoding class services presented in this thesis are designed to handle the visual content adaptation requirements of users with different terminal capabilities and changing network conditions. These services are capable of controlling the bit-rate and provide error-resilience for the video applications. The challenge in designing transcoding class services is how to ensure that such services can provide an efficient performance while operating with minimal delay.

The following subsections introduce the video transcoding services developed in the research work presented in this thesis. These services are performed by a MPEG-4 video transcoder, which is able to apply rate-control, error-resilience, and congestion control services on the input video streams. In the design of this transcoding services the trade-off between the operational-performance and perceptual quality was taken into account. As a result, fast and effective transcoding architecture was build that can be utilised as an active service. The details of how this trade-off was considered and the related implementation details of the transcoder are given in Section 5.3.5.

5.3.1 Transcoding Architecture

Video transcoding is the process of converting the format of an input coded video content into another format [VetrA], [XinJ]. It is primarily used to adapt the bit-rates, temporal and spatial resolutions of the incoming video streams, as well as to provide syntax translations between different video coding standards. Moreover, video transcoding with error resilience properties is particularly popular due to the fixed/wireless inter-networking interoperability issues, and hence addressed in literature [ReyeG], [DogaS], [KimI], [EminS], [ChioH], [XiaM].

The video transcoder presented in this chapter is designed for rate-control and insertion of error-resilience, and is based on the MPEG-4 video coding standard [MPEG4]. It was built using standard compliant MPEG-4 decoder and encoder software which was developed by the Microsoft Corporation and is available for download with a free licence in [Mpeg4Sw]. There are a number of reasons for using MPEG-4 standard in the transcoder. Firstly, MPEG-4 was
originally designed as the multimedia compression standard for mobile multimedia applications which are transmitted over error-prone and band-limited channels. Secondly, 3GPP standardisation body has recommended the use of MPEG-4 video codec for video communications over 3G wireless networks [TS26.235]. In addition, MPEG-4 contains various error-resilience capabilities as its core future, which makes it ideal for transporting a low bit-rate video stream in error-prone environments, such as wireless networks (e.g., GPRS, EGPRS, UMTS etc.). Finally, MPEG-4 is popular in the Internet multimedia applications with primarily targeted bit rates from 56 Kbits/s to 2 Mbps, and error-resilience properties, which makes it suitable to the current and near-term Internet [Mp4].

Multiple-layer video streams are more complex to encode, transcode, and decode. For this reason, the layered video coding schemes, although being scalable, are not widely adopted in today’s applications. In addition, the wireless networks (e.g., GPRS, UMTS) are bandwidth-limited environments with processing power limited end-terminals. In the light of these facts, the transcoder was designed to provide adaptation for base-layer video streams only. The transcoder is only able to process CIF (Common Intermediate Format), QCIF, and SQCIF (Sub-Quarter Common Intermediate Format) video sizes. Considering the targeted usage scenarios (low delay applications over band-limited and error prone channels), it was not necessary to provide compatibility for other video resolutions. However, since the transcoder is mainly targeted for video transmission over bandwidth limited wireless channels, most of the experiments were performed with the QCIF resolution.

In general, transcoders are designed to work either in the Discrete Cosine Transform domain (DCT-domain) or in the pixel domain. Operating in the pixel-domain offers higher flexibility in terms of executing various different services simultaneously (i.e., controlling the output rate while inserting necessary amount of error robustness). Therefore, the transcoding architecture presented in this work was developed to operate in pixel-domain. This architecture is based on the Cascaded Pixel-Domain Transcoder (CPDT) formation. Figure 5.1 shows the block diagram of this transcoder. In this type of transcoding operation the decoder section fully decodes the input video information and the partial-encoder re-encodes the decoded video in the desired format. Typically, CPDT operation is computationally more expensive than the DCT-domain architectures. Consequently, the reduction of the complexity while providing a highly functional and efficient architecture has been the main driving force behind many research activities in this area. Therefore, during the developments stages of the transcoder presented here, a number of methods have been employed for reducing the computational complexity. The details of the complexity reduction techniques and the analysis of the transcoder’s execution performance are discussed in Section 5.3.5.
5.3.1.1 Operation of the Transcoder

The transcoder is designed to produce either I or P pictures, and its output GOP structure is \((x, 1)\) where \(x\) stands of the preferred frequency of I frames between P frames. B-pictures are not used in the transcoder as they slow down the operation speed. Moreover, these pictures are mainly useful for improving the coding efficiency for video storage applications rather than low-latency streaming and real-time conferencing applications.

As can be seen from the Figure 5.1, the transcoder is made of a decoding and an encoding section. The frames of the incoming compressed video stream are decoded sequentially and each decoded video frame data is forwarded to the partial encoder section. The Variable Length Decoding operation (VLD) extracts the texture (quantised DCT coefficients) and motion information (motion vectors) from the compressed bit-stream. This texture information is the transformed and quantised residual signal \((S_{res})\) during the original encoding process.

\[ S_{res} \]

\[ \text{is then inverse quantised } (Q^{-1}) \text{ and inverse transformed } (DCT^{-1}) \text{ to obtain the current-frame’s texture difference information } (S_{diff}) \text{ form the previous frame. This data is then added to the motion compensated (MC) version of the previous-frame } (F’_{prev}) \text{ to construct the complete current-frame } (F’_{curr}). \text{ At this stage, the video information is represented in the pixel-domain and is ready for the encoding process.} \]
In the partial-encoder, the reconstructed picture coming from the decoder is subjected to a simplified encoding operation which does not employ motion-vector estimation or refinement processes. Decoded Motion Vectors (mv) are passed directly to the partial encoder without any interpolation. This is called the motion vector re-use scheme in transcoding, and its use is further elaborated in Section 5.3.5. The encoding process starts with $F_{\text{curr}}$ being subtracted from the motion compensated version of the previously encoded frame ($F'_{\text{prev}}$), which gives the new predication difference signal ($S_{\text{diff}}$). This signal is first transformed by a DCT module. Then, it is re-quantized possibly with a different quantization step size than the decoder section’s quantiser (i.e., if rate-control is used). The quantized DCT coefficients and the motion vectors are multiplexed by a variable length coder and the output bit stream is transmitted. The partial encoder contains a so-called local decoder to reconstruct the encoded frame and store it to be used as $F'_{\text{prev}}$ in the encoding of the next frame. The rate-control and error-resilience functions are executed during this encoding process. The details of these operations are given in Sections 5.3.2 and 5.3.3 respectively.

5.3.1.2 Transcoder Capabilities and its Design Criteria

The transcoder was designed to perform Rate-Control (RC) and Error-Resilience (ER) services on the input video streams. The RC service is generally required to provide better utilization of the bandwidth, rate-matching between heterogeneous networks, and fair treatment to responsive (i.e., TCP-based) applications. In the context of the research work presented in this thesis, the RC service is employed to smooth out the fluctuations in the throughput and convert input high bit-rate streams into lower rates [$E_{\text{min}}S$]. As a result, users with diverse terminal capabilities, personal requirements, and service level agreements can be accommodated. The mechanism employed for rate-control is a macroblock-based rate control algorithm, named as Test Model 5 (TM5) [Tm5]. As shown in Figure 5.1, the RC block regulates the quantization step size of the Partial Encoder in order to compress the input bit-stream at the target bit-rate.

The ER services are essential in alleviating the artefacts resulting from the transmission errors. The transcoder incorporates Data Partitioning (DP), Video Packet Resynchronization (VPR), Header Extension Code (HEC), Adaptive Intra Refresh (AIR), and Scene and Channel Adaptive Intra Refresh (SC-AIR) algorithms as its ER services. While DP, VPR and HEC are applied directly on the quantized video information, AIR or SC-AIR algorithms are executed prior to the encoding process in order to alter the macroblock mode decisions extracted from the decoding operation. Depending on the operator’s choice, either AIR or SC-AIR algorithms can be chosen as the intra update mechanism. As illustrated in Figure 5.1, both algorithms make use of the decoded motion vector information in order to decide which macroblock to update. However, the SC-AIR
The transcoding architecture presented here differs from the existing error-resilient video transcoders in a number of ways. Firstly, the transcoder was designed for use in scenarios that involve low-latency video communications over 3G networks. Therefore, its ER services were specifically designed to be computationally efficient (i.e., introduce minimal latency) and provide robustness against channel errors. Other than the standard MPEG-4 error-resilience tools, the transcoder employs a scene-activity and channel adaptive intra-refresh algorithm. This algorithm can dynamically analyse the input video content and provide the best possible error-resilience given the motion activity of the video input and the instantaneous channel condition.

The trade-offs between the operational-performance and perceptual video quality was the main design criteria in the development of the transcoder’s services. For instance, no motion re-estimation or refinement method was employed in order not to sacrifice from the delay performance. Nevertheless, previous research have shown that such methods can yield only a little video quality gain in the case of transmission over an error-prone wireless channel (e.g., 2.5G channel) [SadkA]. Similarly, various other complexity reduction techniques have been used to simplify the transcoder design. An elaborate discussion of these techniques is given in Section 5.3.5.

5.3.2 Rate-Control Service

RC service is the transcoding operation that is used to smooth out the fluctuations in the throughput and convert input high bit-rate streams into lower rates. The objective of this operation is to reduce the bit-rate while maintaining low complexity and achieving the highest quality possible [VetrA]. By this way, users with diverse terminal capabilities, personal requirements, and service level agreements can be accommodated effectively. RC service could also be utilised to ease the congestion on the network links. In the case of congestion, applications may suffer from uncontrolled packet losses due to the dropping mechanism employed at the network routers. However, in an active networking scenario, congestion may be alleviated by intelligent packet discard mechanisms used together with RC service. Video streams could be transcoded into lower rates to reduce the router buffer occupancy and consequently the probability of loosing a video packet.

RC transcoding can be performed in either DCT-domain [KeesG], [MorrD], [AssuP] or pixel-domain [SunH], [KeesG], [YounJ], [AssuP2]. Although DCT-domain approach is less complex,
pixel-domain approach offers better flexibility in service provision. In other words, the transcoder architecture presented here is not merely a bit-rate regulator but is also capable of performing various other services. The RC service is only one of the transcoding class services and is usually used together with ER services. Provision of such functionality makes the use of pixel-domain transcoder architecture mandatory.

5.3.2.1 Rate-Control Algorithm

The mechanism employed for rate control is version of a macroblock-based rate control algorithm, named as TM5 [Tm5]. The algorithm presented here is simplified by omitting the B-frame related calculations. This algorithm adjusts the quantisation step size per macroblock basis and does not employ any frame skipping technique. The rate-control algorithm's operation can be described in three steps which are:

1. Bit allocation
2. Rate Control
3. Adaptive Quantisation

In the bit allocation step, based on the determined output bit-rate, each GOP is assigned a target number of bits. The algorithm tries to meet this target by calculating the required quantisation step size for each macroblock of a frame. The first procedure in the bit allocation step is to estimate the number of bits available to code the next picture. After a picture of a certain type (I or P) is encoded, the respective global complexity measure \(X_i\) or \(X_p\) is updated. Complexity measure is related with the spatial variation in a picture. The encoder calculates a different complexity measure \(X\) for each class of coded picture. The complexity measure for I and P frames are calculated as follows:

\[
X_i = S_i \times Q_i \tag{5.1}
\]

\[
X_p = S_p \times Q_p \tag{5.2}
\]

where \(X_i\) is the complexity measure for the I-frames, and \(X_p\) is for the P-frames. \(Q\) is the average quantization step size for the whole picture and \(S\) is the number of bits generated in the encoding of the current picture. For the first I and P frames in a GOP, the initial complexity values can be calculated using the following formulas:

\[
X_i = \frac{(160 \times \text{bit-rate})}{115} \tag{5.3}
\]

\[
X_p = \frac{(60 \times \text{bit-rate})}{115} \tag{5.4}
\]

A high value of the complexity measure indicates a coded picture that contains a significant amount of spatial detail. Having obtained the complexity values, a target bit rate \(T\) is determined
for the next picture to be coded based on the complexity measure for that frame type (I or P) and on the number of bits reserved for the GOP.

\[
T_i = \max \left\{ \frac{R}{N_p X_p}, \frac{\text{target bit-rate}}{1+\frac{X_p}{X_i X_p}} \times 8 \times \text{picture rate} \right\}
\] (5.5)

\[
T_p = \max \left\{ \frac{R}{N_p}, \frac{\text{target bit-rate}}{8 \times \text{picture rate}} \right\}
\] (5.6)

where \( R \) is the number of bits assigned to a GOP and after the encoding of the current frame it is updated as:

\[
R = R - S_{\text{lp}}
\] (5.7)

Before encoding the first picture in a GOP,

\[
R = G + R
\] (5.8)

where,

\[
G = \text{bit-rate} \times N / \text{picture rate}
\] (5.9)

and \( N \) is the number of pictures in the GOP. At the start of the sequence \( R \) is zero. \( N_p \) is the number of P-pictures remaining in the current GOP in the encoding order.

In the rate control step, the fullness of the virtual buffers \( d^i \) and \( d^p \) (for I and P frames) are computed, one for each class of coded picture. As each macroblock is encoded, the fullness of the appropriate virtual buffer is calculated according to the number of bits already coded for the current picture and the target bit-rate \( T \) for the current picture. This fullness of the virtual buffer is then fed back to the quantization control and is used to choose the quantization step size for the current macroblock. The fullness of the virtual buffers at macroblock \( j \) of each picture type is calculated with the following equations:

\[
d^i_j = d^i_0 + B_{j-1} - \frac{T_i \times (j-1)}{\text{macroblock_count}}
\] (5.10)

\[
d^p_j = d^p_0 + B_{j-1} - \frac{T_p \times (j-1)}{\text{macroblock_count}}
\] (5.11)
where $B_j$ is the number of bits generated by encoding all macroblocks in the picture up to and including $j$. The final fullness of the virtual buffers, where $j = macroblock\_count$, are used as the initial $d_0^i$ values in encoding the next picture of the same type. At the onset of a GOP, the initial values for $d_0^i$ are calculated as:

$$d_0^i = d_0^p = 20 \times \frac{bit\_rate}{picture\_rate}$$  \hspace{5cm} (5.12)

The virtual buffer fullness is then used to compute the reference quantization parameter $Q_j$ for every macroblock.

$$Q_j = \left( \frac{d_j \times 31}{2 \times \frac{bit\_rate}{picture\_rate}} \right)$$ \hspace{5cm} (5.13)

Finally, the chosen quantization step size is modulated according to the spatial activity in the macroblock. The spatial activity measure for the macroblock $j$ is calculated from the four luminance frame-organised sub-blocks ($n=1..4$) and the four luminance field-organised sub-blocks ($n=5..8$) using the original pixel values.

$$act_j = 1 + \min(vblk_1, vblk_2, vblk_3, ..., vblk_8)$$ \hspace{5cm} (5.14)

where,

$$vblk_n = \frac{1}{64} \times \sum_{k=1}^{64} (p_k^n - p_{\text{mean}}^n)^2$$ \hspace{5cm} (5.15)

$$p_{\text{mean}}^n = \frac{1}{64} \times \sum_{k=1}^{64} p_k^n$$ \hspace{5cm} (5.16)

and $p_k$ are the sample values in the $n^{th}$ original 8x8 block. Using these, the normalised activity measurement is calculated as:

$$nact_j = \frac{2 \times act_j + \text{avg\_act}}{act_j + 2 \times \text{avg\_act}}$$ \hspace{5cm} (5.17)

In this equation, the $\text{avg\_act}$ is the average value of $act_j$ in the last encoded picture. On the first picture, $\text{avg\_act} = 400$. If the variance in the luminance content of the macroblock (i.e., the spatial variation and detail) is higher than average, then the quantization step size is increased. The final estimated quantisation step size for a particular macroblock is given as:

$$mquant_j = Q_j \times nact_j$$ \hspace{5cm} (5.18)
This means that areas that contain high detail (i.e., highly motion-active areas) are quantized more coarsely. This rate control scheme keeps a close control over the output rate and aims to minimize the variation in the contents of the output buffer. Quantization is controlled on a macroblock by macroblock basis, which means that the TM5 algorithm can react quickly to changes in picture content and encoded bit-rate. Nevertheless, this also means that the visual quality of the decoded video sequence is changeable within a single frame. The final value of $mquant_j$ is clipped to the quantiser range (1-31) and used as the required quantisation parameter for the current macroblock.

### 5.3.2.2 Rate-Control Service Performance Tests

A number of tests were conducted to demonstrate the effects of applying rate-control on the input Variable Bit-Rate (VBR) video stream. The TM5 algorithm implemented on the transcoder is able to produce a Constant Bit-Rate (CBR) stream at the output of the transcoder. Conversion of a higher quality VBR video stream into a lower quality CBR one requires re-quantisation of the decoded stream with variable quantisation step sizes such that the transcoded video stream does not exceed or fall below the target bit-rate. The reduction in the output bit-rate results in degradation and fluctuations in the decoded video quality. In the following subsections, the effects applying rate-control service at the transcoder are illustrated and discussed.

**Experimental Procedure**

Two standard video sequences, namely the “Foreman” and “Students”, were chosen for the experiments performed to demonstrate the performance of the rate-control services. These two sequences were used since they represent two distinctively different motion activity scenarios. While the “Foreman” sequence is highly motion-active, the “Students” sequence is an activity limited sequence. Considering that the rate-control algorithm is not able to compress the motion information, it is interesting to compare its performance with two sequences of highly different motion activity scenes.

To test the rate-control algorithm, an MPEG-4 video encoder was used. Both video sequences at QCIF (i.e., $176 \times 144$ pixels) resolution were encoded with a fixed quantisation step-size ($Q_p = 2$). The encoder was set to produce a base-layer stream with no rate control. The encoding operation modes were set to 10fps (frames per second) and I-P-P-P-P-... layout for both video sequences. The duration of the encoding operation was limited to 13 seconds (i.e., equivalent of 130 frames at 10fps encoding rate). This resulted in an average bit-rate of 445kbit/s for the “Foreman” stream and 230kb/s for the “Students” stream. These streams were then transcoded using the rate-control service at a number of different bit-rates (i.e., 128, 96, 64, 32 kbits/s). The transcoded streams were then decoded for measuring the effects of the RC service on the video quality.
Results and Discussion

Three different tests were performed on the RC serviced streams. The first test involved measuring the transcoder’s throughput for every second at different target bit-rates. The second test involved quantifying the quality degradation of the decoded video stream due to the application of the RC service. The results were established in the form of PSNR measurements. Finally, subjective tests were performed to visualise the degradation and fluctuations in the video quality.

The performance of the transcoder in terms of producing CBR streams for both “Foreman” and “Students” sequences are shown in Figure 5.2 and Figure 5.3 respectively. As can be seen from these figures, the transcoder is able to smooth the rate fluctuations in the input stream and produce a constant bit-rate output. However, the rate-control algorithm does not to smooth the peaks occurring due to the I-frames (e.g., as in the first second of both streams). When an I-frame is used, the rate-control algorithm tries to maintain a balance between the future P-frames and the I-frame in terms of the number of bits allocated for the encoding process. In other words, an I-frame should be encoded with enough number of bits such that the future P-frames do not get degraded due to the use of considerably low quality first reference frame in the motion compensation process. For this purpose, the RC algorithm accounts for the increased number of bits used for the encoding of the I-frame and regulates the number of bits that should be used for the P-frames accordingly such that the target bit rate is achieved at the end of the GOP.

Although the RC algorithm tries to match the target bit-rate for every second of the video stream (i.e., target bit-rate is given in number of bits per second), this may not be possible if there are sudden and large scale object movements or scene changes in the video stream. This can be observed on and after the 10th second of the “Foreman” stream, where a sudden camera movement results in a complete scene change (see Figure 5.2). At around the 10th second, the camera turns quickly towards another direction, causing many macroblocks with relatively larger motion vectors to be encoded. This causes a sudden increase in the bit-rate and the RC algorithm tries to compensate this increase by lowering the number of bits allocated for the future frames in the GOP. As can be seen on Figure 5.2, the fluctuation on the bit-rate is stabilised by the end of the 12th second. However, this is not the case for the RC service at 32kbits/s. At this rate and for this particular video stream, the algorithm fails to match the target bit-rate even with the largest quantisation step size since no frame dropping technique is applied. On the 10th second, the algorithm increases the quantisation step size to its highest value (i.e., $Q_p = 31$), yet is unable to cope with the sudden increase in the bit-rate. Consequently, it minimises the number of available bits for the future frames and fails to stabilise at around the target bit-rate for every second.
Chapter 5. Active Services QoS Support

Figure 5.2: Effects of the RC Service on the Transcoder’s Throughput for the “Foreman” Stream

Unlike the case with the “Foreman” sequence, the RC service enabled transcoding of the “Students” stream produces a very smooth output since it does not involve scene changes or sudden movement of video objects. Therefore, except for the first second where an I-frame exists, the target bit-rate is matched at every second. The performance of the RC algorithm on the “Students” stream can be seen in Figure 5.3. The use of rate-control has a significant effect on the decoded video quality. Since the RC algorithm cannot compress the motion vector information, the perceptual quality attainable in any frame is inversely proportional with the total number of bits required to encode the motion information of the same frame. Moreover, differences in the level of residual texture information between the frames mean that each frame is subject to different level of compression (i.e., different average Qp). Therefore, applying the RC service on the input VBR streams creates fluctuations in the perceived video quality. In TM5, such fluctuations become perceptually more noticeable for video sequences with changing motion-activity levels and/or scene changes. In Figure 5.4 and Figure 5.5, the effects of applying RC service at different bit-rates on both streams are depicted. The average PSNR values of each RC service at 128kbits/s, 96kbits/s, 64kbits/s, and 32kbits/s for the “Foreman” and the “Students” sequences are given in Table 5.1.
Figure 5.3: Effects of the RC Service on the Transcoder’s Throughput for the “Students” Stream

As can be observed from Figure 5.4, the recorded PSNR values of the “Foreman” stream fluctuate considerably between the 68th-110th frames, indicating variation in the perceived video quality. As explained earlier, after the 10th second, the RC algorithm operating at 32kbits/s cannot efficiently
cope with the sudden increase in the bit-rate, which explains the pattern difference in the PSNR plot, particularly after the 110th frame.

The "Students" stream responds differently to the RC algorithm. Its lower PSNR values are obtained during the first second of the transcoded video stream. During the experiments it was recorded that the first frame of the "Students" stream contains around 16% more texture bits on its I-frame (first frame) than the I-frame of the "Foreman" stream. In addition, because of the low motion-activity of this sequence, the TM5 algorithm is able to spare more bits on the texture information of the following P-frames. For this reason, the PSNR values recorded during the first second starts low but gradually increase and after the first couple of frames steady PSNR patterns are observed. This shows that the decoded video quality compared to the foreman sequence is less fluctuating and therefore perceptually less disturbing.

The RC algorithm's performance was also observed with subjective tests. Figure 5.6 and Figure 5.7 depict sample frames form the RC service transcoded "Foreman" and "Students" streams respectively. In both of these figures, the quality difference between the transcoding operations at different bit-rates can be observed.
Chapter 5. Active Services QoS Support

<table>
<thead>
<tr>
<th>Avg. PSNR</th>
<th>RC Service</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>128kbits/s</td>
</tr>
<tr>
<td>Foreman</td>
<td>34.26dB</td>
</tr>
<tr>
<td>Students</td>
<td>36.58dB</td>
</tr>
</tbody>
</table>

Table 5.1: Average PSNR Values for the RC Service

However, the most notable quality difference occurs between the 64kbits/s and 32kbits/s transcoded video streams; a finding in agreement with the PSNR results given in Figure 5.4 and Figure 5.5. However, this is not surprising considering that a bit-rate of 32kbits/s is the half of 64kbits/s. A similar level of quality difference exists between 128kbits/s and the 64kbits/s as can be computed from Table 5.1, and seen in Figure 5.6 and Figure 5.7. The justification of this argument can be made with reference to the operation of the TM5 algorithm. Considering the TM5 equations introduced in Section 5.3.2.1, it can be realised that the target bit-rate is a multiplicative factor in the calculations made by the algorithm and therefore a rate-reduction from 128kbits/s to 64kbits/s or 64kbits/s to 32kbits/s halves the number of bits that can be allocated for the encoding of the frames in a GOP.
Figure 5.6: Subjective Test Results of the 2\textsuperscript{nd}, 36\textsuperscript{th}, 96\textsuperscript{th}, and 133\textsuperscript{rd} Frames of the Transcoded “Foreman” Stream: (a) VBR at avg. 445kbits/s, (b) with RC = 445kbits/s, (c) with RC = 128kbits/s, (d) with RC = 96kbits/s, (e) with RC = 64kbits/s, (f) with RC = 32kbits/s
Figure 5.7: Subjective Test Results of the 2nd, 38th, 86th, and 133rd Frames of the Transcoded "Students" Stream: (a) VBR at avg. 230kbits/s, (b) with RC = 230kbits/s, (c) with RC = 128kbits/s, (d) with RC = 96kbits/s, (e) with RC = 64kbits/s, (f) with RC = 32kbits/s
5.3.3 Error-Resilience Services

In addition to providing rate-control functionality in the networks, error-resilience tools are also required to provide protection for the video streams prior to their transmission. Noisy channel conditions (e.g., as in wireless channels) and packet losses due to congestion (e.g., as in the Internet) introduce errors in the compressed video streams, which are manifested as degradations in the perceived quality of the decoded video. Due to the predictive coding techniques applied in MPEG-4, errors are likely to propagate into the future frames, making video communications perceptually unacceptable. To mitigate such effects, MPEG-4 has adopted a number of error-resilience tools that makes the video streams more robust against error-prone transmission channels [TallR] [MPEG4].

However, there exists a plethora of video applications all of which may not be optimised for error-resilient transmission. This is due to the fact that most error-resilience tools reduce coding efficiency, waste bandwidth, and increase the encoding and decoding complexities. Therefore, when and where necessary, users should be provided with error-resilience services by the intermediate points (i.e., active nodes) in the networks. The problems addressed above call for error-resilience transcoding units to be deployed in the networks. In this section, the ER services of the RC service capable transcoding architecture are presented.

In the literature, several error-resilience transcoding architectures have been proposed [DogaS], [ReyeG]. The common focus of these studies was to develop transcoding architectures that would incorporate certain error-resilience measures that can be injected into the video streams at the static and intermediate network nodes (i.e., media gateways or proxies). However, to the knowledge of the author of this thesis, no error-resilient transcoding architectures have been developed within scope of active networking research. Therefore, the error-resilience (ER) services presented in this section have been designed considering the fundamental properties of the active services and their requirements in terms of complexity. That is to say, considering the media processing load needed to be handled by the active nodes, active services are required to be lightweight software modules that are optimised to make use of minimal node resources. The proposed transcoding architecture utilises computationally simple but effective ER services. The computational simplicity refers to the amount of transcoding delay introduced by the introduction of a new service. If this service does not hinder the real time operation of the transcoder, then it is considered to be a computationally simple algorithm. These services are:

- Video Packet Resynchronisation
- Header Extension Code
- Data Partitioning
5.3.3.1 Video Packet Resynchronisation (VPR)

VPR is the process of inserting frequent and periodic resynchronisation markers into the bitstream by dividing it into video-packets [MPEG4]. In the case of an error in the video stream, the decoder may lose synchronisation (i.e., become unable to identify the exact location of currently decoded data in the frame where it belongs). Unless any compensatory measures are implemented, loss of synchronisation results in information loss in the current frame (sometimes loss of the whole frame), and causes the quality of the decoded video to degrade rapidly due to accumulating prediction errors. By employing the VPR technique, the adverse effects of synchronisation loss can be considerably limited.

A video packet resynchronisation marker is a 17 bit codeword and indicates the beginning of new video packet. Each video packet starts with it, continues with the macroblock number, quantisation parameter, and the motion vector and the DCT data for the first macroblock. The same order (except the resynchronisation marker) is followed for the consecutive macroblocks as long as the predetermined video packet size is not exceeded. In some cases, as explained in Section 5.3.3.2, a header extension code is encoded between the macroblock number and the quantisation parameter fields. A typical structure of a video packet is depicted in Figure 5.8. It should be noted that the ordering of the video packet contents is different if data partitioning mode is enabled. This case is further explained in Section 5.3.3.3.

![Figure 5.8: The Structure of a Video Packet](image)

<table>
<thead>
<tr>
<th>Resynch. Marker</th>
<th>MB_n no.</th>
<th>QP_n</th>
<th>HEC</th>
<th>MV_n &amp; DCT_n data</th>
<th>MB_m1 no.</th>
<th>QP_m1</th>
<th>MV_m1 &amp; DCT_m1 data</th>
</tr>
</thead>
</table>

**MB no.**: Macroblock Position in the Frame  
**QP**: Quantisation Parameter of the Macroblock  
**HEC**: Header Extension Code  
**MV**: Motion Vector Information of the Macroblock  
**DCT**: Quantised DCT Coefficients of the Macroblock

A video packet does not contain a specific number of macroblocks, instead its size is defined in terms of bits it contains. A possible distribution of video packets in a video frame is shown in Figure 5.9. It can be observed from this figure that the video packets do not need to contain the same number of macroblocks, yet their sizes in bits should all be similar. In [TallR] it was stated that the recommended size of MPEG-4 video packets should be based on the output video bit-
rates. For instance, for 24kbits/s it is recommended to set the video packet size to 480 bits; and 736 bits is recommended for data rates between 25kbit/s and 48kbits/s. Other studies have related the optimum video packet size to the wireless channel conditions as well as the bit rate [WorrS2], [SoarL]. Preliminary experiments performed with the transcoder had shown that 700 bits was a suitable value as the video packet length for the experimental scenarios in consideration.

![Figure 5.9: Distribution of Video Packets over a Video Frame](image)

VPR also involves modification of motion vectors in order to remove the data dependencies between the video packets of the same frame. Thus, when an error is detected, the decoder can localise this error within the video packet and proceed to the next resynchronisation marker. This helps the decoder continue decoding without discarding all the data before the next synchronisation codeword which indicates the next video packet or a new frame. Overall, VPR method is a simple but useful error resilience mechanism against the channel errors. The VPR performance is illustrated with the experiments presented in Section 5.3.3.6.

### 5.3.3.2 Header Extension Code (HEC)

HEC provides enhanced resilience against the corruption of frame header due to channel errors [MPEG4]. HEC is a single bit field encoded in the video packet, and indicates whether the frame header is repeated in the packet header. If this bit is set, then the decoder can compare the repeated header information against the frame header. When a corrupted frame header is detected, the decoder can use the HEC related information to continue decoding the rest of the frame data instead of discarding it completely. The video packets which contain the HEC related header information can also be decoded independently. The necessary information to decode a video packet is included in the header extension code field, provided that the HEC is set to 1.

Utilising HEC can improve robustness against frame losses since it helps the decoder to detect and correct the corrupted information in the frame headers. However, the use of HEC reduces compression efficiency due to the repeated header fields. In the design of the transcoder it was
decided to enable the redundant HEC information only once in the second video packet of every video frame.

**5.3.3.3 Data Partitioning (DP)**

VPR is helpful in confining errors to individual video packets. When an error occurs inside the video packet, the video decoder is unable to determine whether the motion or texture information is affected. Thus, the decoder discards all the erroneous macroblocks and performs error concealment by replacing the discarded macroblocks with the corresponding blocks from the previous frame. This operation introduces prediction errors in the decoded video due to the possible differences in motion and texture information between the corresponding macroblocks of the consecutive frames. However, if the decoder could determine that the error is not on the motion data, it could salvage the motion vectors and utilise them in the motion compensation process.

DP tool in MPEG-4 alleviates this problem by providing a clear separation between the motion and texture information of a video packet using a motion boundary marker (MBM) [MPEG4]. MBM is a uniquely decidable code word which marks the end of the motion data and the beginning of the texture data. The organisation of video and texture information is shown in Figure 5.10. As it can be seen from this figure, all the motion-related information (for all macroblocks contained in a video packet) are placed in the motion partition, and all those that relate to texture (i.e., DCT data) are placed in the DCT data partition.

![Figure 5.10: The Structure of a Video Packet in Data Partitioning Mode](image)

If an error occurs in the motion partition, the decoder would then discard all the macroblock data inside the video packet by marking the macroblocks as skipped. An error affecting the texture partition on the other hand, would allow the decoder to use motion vectors to perform motion compensation.

DP service is dependent on the VPR service in order to operate. Therefore, the transcoder operates the DP and VPR services together. The analysis of DP performance and the details on the combinational usage of different ER services are given in Section 5.3.3.5. An analysis about the delay performance of this service is given in Section 5.3.6.
The performance of the DP service could be enhanced if unequal-error-protection (UEP) schemes are used to provide weighted protection to different partitions of a video packet. In theory, giving higher protection to the motion partition may yield better video quality in error-prone channel. Study on any UEP scheme was not carried out in the research work presented here and is regarded as a subject of the future investigation.

5.3.3.4 Adaptive Intra-Refresh (AIR)

The aforementioned services can be considered to be preventive mechanisms since their purpose is to limit the effects of channel errors on the video quality. Intra-refresh algorithms are different category of error resilience services since they aim at recovering the already corrupted video information and stopping it from propagating into the future frames. In that sense, such algorithms can provide considerable performance improvement on the corrupted video streams. One basic but effective intra-refresh algorithm, namely the adaptive intra-refresh (AIR), is given in [MPEG4]. AIR algorithm is an integral part of the error-resilient transcoding architecture presented in this thesis. This algorithm is particularly useful in channel conditions where the occurrence of errors have a bursty nature (i.e., as in the 3G networks).

Lost or corrupted video information causes errors, which are perceptually disruptive for the users. Due to the nature of the predictive coding used in the block-based video coders (e.g., MPEG-4, H.263 etc.) an error occurring in the reference frame tends to propagate to the predicted future frames. While errors due to packet loss are more associated with the fixed networks (e.g., the Internet), wireless channels are known to introduce noise into the transmitted video signal, which may cause the decoder to misinterpret the received data and lose synchronisation. AIR is an effective error resilience tool for recovering from the adverse effects of the corrupted or lost video packets. It is a low complexity algorithm and only requires utilising the readily available motion information in the frame being encoded.

The AIR algorithm is responsible for sequential introduction of a certain number of intra-coded macroblocks in the P-frames of the video stream. By this way, the lost or corrupted video information can be refreshed and the propagation of errors into the future frames is prevented. In other words, AIR is firstly a recovery mechanism from the adverse effects of the channel errors and secondly a preventive mechanism for the propagation of these errors.

As the refresh rate of the AIR algorithm is increased, the recovery from the errors speeds up but the compression efficiency may be sacrificed. Choosing an efficient AIR rate requires a priori knowledge about the motion levels and the resolution of the video sequence, as well as the transmission channel characteristics in order to determine the required number of intra blocks.
which would optimise the perceived video quality. While inadequate number of AIR blocks may be deficient in refreshing the errors fast enough, excessive number of AIR blocks may result in increased quantisation distortion especially if the encoding process is rate controlled. Traditionally, the number of AIR blocks inserted in frames is fixed and chosen before the encoding process [MPEG4]. In this algorithm, a motion map is formed to mark the location of the macroblocks with motion. The AIR algorithm starts from the top of the frame and refreshes a given number of macroblocks in the motion map. Once a macroblock is refreshed, it is marked as inter block in the motion map so that it is not refreshed again before all other macroblocks in the motion map are refreshed at least once. When the refresh algorithm points at the last macroblock of the frame, the AIR pointer is reset to the beginning of the frame. The operation of the standard MPEG-4 AIR algorithm is illustrated in Figure 5.11.

![Figure 5.11: Standard MPEG-4 AIR Algorithm (1 = motion, 0 = no motion) [MPEG4]](image)

5.3.3.5 Scene and Channel Adaptive Intra-Refresh (SC-AIR)

SC-AIR service is a novel intra-refresh algorithm whose operation is similar to AIR. However, this algorithm was developed by the author of this thesis to account for the deficiencies of the AIR algorithm in terms of providing adaptation for the scene activity and changing channel conditions. A comprehensive presentation of this algorithm together with its performance analysis is given in Section 5.3.4.

5.3.3.6 Error-Resilience Services Performance Tests

Error-resilience services are essential for providing an acceptable quality of service quality for video transmission over error-prone channels. Therefore, a number of tests were performed to demonstrate the effects of introducing ER services in the networks. These tests involved the execution of such services on the non-resilient video streams traversing between different networks. Introduction of ER services is shown to alleviate the severity of quality degradation resulting from the transmission errors. In the following subsections, firstly, the experimental setup
and scenario are explained. Secondly, the experimental results and a discussion about the implications of these results are given.

**Experimental Procedure**

To illustrate the performance of the ER services, a series of experiments were conducted using the simulation scenario shown in Figure 5.12. In this scenario, the video source lies in the fixed network and transmits live or stored video content to its subscribers in the 3G network which uses W-CDMA technology in its radio access network. Upon the request of a mobile user, the video server sends the video stream through an active node, which is installed at an appropriate location between two networks. This node executes the transcoding module and is assumed to be aware of the user profile, requirements and the channel conditions. Consequently, it is capable of modifying the incoming bit-stream to the required format. Mobile users in the 3G network experience varying channel conditions due to their location, speed and various interference sources, as represented by the $\beta(t)$ function. The transcoding operation was implemented by using the C/C++ programming language, and the wireless channel effects were emulated using the W-CDMA channel simulator [KodiC1].

In the experiments, the “Students” and the “Foreman” sequences were used. As explained in Section 5.3.2.1, they can be regarded as low and high motion-active sequences respectively. The purpose of using such contrast in the experiments is to identify the performance differences of the proposed ER services on characteristically different video streams.

The video server depicted in Figure 5.12 employs a MPEG-4 video encoder and is set to produce a single base-layer stream with no rate control (with a fixed quantisation step size at $Q_p = 2$). The
Chapter 5. Active Services QoS Support

server produces QCIF sized video streams encoded at 10fps, and with I-P-P-P-P-.... layout for both video sequences. The duration of the encoding operation was limited to 13.4 seconds (i.e., 134 frames) for both video sequences. The error-resilience options of the encoder were turned off as the link between the source and the transcoder is assumed to be lossless and error free.

In the experiments performed, the transcoder was configured to convert the incoming variable rate bit-stream (i.e., on average at 230kbits/s for the “Students” and 445kbits/s for the “Foreman”) to a constant bit-rate output at 128kbits/s. The output transmission rate was chosen to be 128kbits/s because it is adequate to provide reasonable perceptual quality for QCIF sized high and low motion-active video streams. The robustness against the adverse effects of wireless channel errors was enabled by the execution of active services such as DP, VPR and HEC with a fixed packet size of 700bits, and the standard AIR. It can be argued that, if a fixed number of AIR blocks are used for every frame, a video packet size of 700bits is a reasonable value that can be used for most video sequences given the 128kbits/s transcoder output and varying channel conditions [EminS], [SoarL]. Similarly, based on the figures presented in [SoarL] and the preliminary study performed, an AIR frequency of 10 blocks was decided to be used in these experiments.

The input video stream was transcoded for a number of times, each time with different transcoder settings. As the first step, the incoming stream was rate-controlled at 128kbits/s and then decoded without introducing any channel errors. This is used for comparing the quality achieved by employing a specific combination of ER services. The next step was to produce a transcoder output which was not subjected to any of the ER services. The performance measurement obtained from such video stream represents the worst possible video quality. Between these two extremes, ER services (i.e., VPR, HEC, DP, and AIR) were progressively applied to other transcoded streams of the incoming video streams. The SC-AIR algorithm is not discussed here since its comprehensive description and analysis are given in Section 5.3.4.

The effects W-CDMA physical link layer on the transcoder’s downlink was simulated by corrupting the transcoded video streams with appropriate error patterns. These error patterns were produced by the W-CDMA physical link layer simulator, which was developed in CCSR [KodiC1]. They are used for emulating the downlink channel conditions for a specific E_b/N_0 (Bit-Energy-To-Noise), channel coding scheme, Spreading Factor (SF), propagation environments, mobile speed, and power control availability.

In these experiments, because of the transcoder’s target output bit-rate, the SF was determined as 16. In addition, the channel coding scheme was set to 1/3 convolutional code (CC 1/3). Since no power control was assumed, Vehicular-A propagation environment at 50km/h mobile speed was
chosen. The video performance at this environment is similar to Pedestrian B environment with power control [KodiC2]. The channel $E_b/N_0$ was selected as 9dB. Given these conditions, the $E_b/N_0$ value of 9dB represent channel Bit-Error-Rate (BER) of approximately $6 \times 10^{-2}$ [KodiC1].

Five different tests were carried out for each video sequence. These tests involved progressive application of different levels of ER services on the 128kbits/s rate-control serviced video streams. Firstly, a rate-control serviced video stream that is not serviced with any of the error-resilience is transmitted over the error-prone wireless link. Secondly, a VPR and HEC serviced version of the same stream is tested. This was followed by the repetition of the same experiment with the addition of the DP service. Finally, the full resilience was provided by adding the AIR service support on top of all other services. In the preliminary work performed, it was observed that 25 simulation runs for each test are adequate for correct representation of the overall channel effects on the video quality. In other words, each transcoder’s output was corrupted with channel errors for 25 times with the same set of seeds for each different test. The corrupted video streams were then decoded and the resulting video quality was measured. The PSNR results of each test were averaged to draw an overall picture of the service performance. The obtained results are discussed in the next section.

Results and Discussion

Exhaustive simulations were run to test the performance of the ER services. Having performed these simulations according to the experimental procedure described in the previous section, their results need to be analysed. The quality of the decoded video is assessed using PSNR measurements and subjective tests.

The PSNR measurements of all five error-resilience tests were plotted on the same graph in order to show the relative performances of different combinations of services. This procedure is applied to both “Foreman” and the “Students” sequences. Figure 5.13 and Figure 5.14 show the PSNR results of the experiments performed using these two sequences. The average PSNR values for all of these tests are tabulated in Table 5.2. Subjective tests were also performed to assess the performance of the ER services. Figure 5.15 and Figure 5.16 illustrate the change in picture quality with the application of different combinations of error-resilience services. Error-free (Figure 5.15 (a) and Figure 5.16 (a)) and error-prone (Figure 5.15 (e) and Figure 5.16 (e)) versions of the same streams are also included in these illustrations in order to provide the upper and lower baselines for the performances of the services.
The PSNR results reveal some interesting findings. As can be seen in both Figure 5.13 and Figure 5.14, the use of combined VPR and HEC services can provide better video quality than the non-error-resilient stream (e.g., no services applied). The use of these services improves the results of the PSNR measurements on average by 0.62dB for the “Foreman” and 0.64dB for the “Students” streams.

<table>
<thead>
<tr>
<th>Avg. PSNR</th>
<th>RC Service</th>
<th>No Service</th>
<th>VPR+HEC</th>
<th>VPR+HEC+DP</th>
<th>VPR+HEC+DP+10AIR</th>
<th>Error-Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foreman</td>
<td>25.67dB</td>
<td>26.29dB</td>
<td>28.45dB</td>
<td>31.33dB</td>
<td>33.98dB</td>
<td></td>
</tr>
<tr>
<td>Students</td>
<td>26.57dB</td>
<td>27.21dB</td>
<td>33.24dB</td>
<td>34.02dB</td>
<td>36.35dB</td>
<td></td>
</tr>
</tbody>
</table>

Table 5.2: Average PSNR Values for the ER Services

In fact, for the given source rate, channel condition, and the propagation environment, $E_b/N_0$ value of 9dB can be considered to represent a moderate channel condition. Therefore, the performance gain obtained from using the combined VPR and HEC service is expected to become more distinctive as the channel conditions worsen (i.e., as $E_b/N_0$ value reduces).

For the “Students” stream, the PSNR plot of this service shows a steady descent and follows a similar pattern to the performance of the decoded non-error-resilient stream. This descent is due to the accumulation of errors in the P-frames. However in the “Foreman” case, a sudden peak in the PSNR plots is observed for both serviced and no-service cases. This is due to the scene change taking place in the “Foreman” sequence between the 95th and 105th frames. In order to cope with this change, the MPEG-4 encoder inserts a number of intra coded blocks in the frames. On average 8.2% of the macroblocks are coded in intra mode during this interval. As a result, the some of the erroneous inter coded macroblocks are refreshed with intra coded ones, which explain the sudden rise in the PSNR measurements.

An illustration of the subjective test results for the VPR and HEC service is given in Figure 5.15 (d) and 5.16 (d). The visual quality improvement achieved in the “Foreman” stream by using this service combination is noticeable when Figure 5.15 (d) is compared to Figure 5.15 (e). However, there is no such perceptual difference noticeable from the comparison of 5.16 (d) to Figure 5.16 (e). Nevertheless, based on these observations it can be claimed that the level of quality attainable by using VPR and HEC services alone is not perceptually satisfactory.
The use of the DP service, on the other hand, brings in a considerable performance gain in the decoded video quality. However, it should be noted that the DP service is always used with the VPR service (as explained in Section 5.3.3.3), and optionally together with the HEC service as well. Here, a combined DP, VPR and HEC service is utilised for providing robust transmission of the video signals over the error-prone W-CDMA channel. The addition of the DP service reduces the amount of video information discarded by the decoder in the case of errors affecting the texture information. As can be observed from the Figure 5.13 and Figure 5.14, the DP service together with the VPR and HEC services enables an improved PSNR performance on average by 2.16dB for the "Foreman" and 6.03dB for the "Students" streams. The difference in the performance gains is due to the motion activity differences between these sequences. For the bitrate of 128kbits/s at the transcoder's output, only 3.1% of the total compressed bits are motion information for the "Students" stream while this figure is 11.3% for the "Foreman" stream. This means that the likelihood of channel errors corrupting the motion information in the "Foreman" stream is considerably higher than the "Students" stream. As explained previously, the DP service is not capable of salvaging the video packet contents when an error occurs in the motion partition of this video packet. The combined DP, VPR and HEC service results in an average PSNR figure of 33.24dB for the "Students" stream (i.e., around 3.11dB less than the error-free performance), and 28.45dB for the "Foreman" stream (i.e., around 5.53dB less than the error-free performance).

The subjective test results for the combined VPR, HEC and DP service are also in line with the PSNR measurements. Compared to the performance of the VPR and HEC services (Figure 5.15 (d) and Figure 5.16 (d)), the utilisation of the DP service resulted in significant improvement on the perceptual quality. Figure 5.15 (c) and Figure 5.16 (c) illustrate this effect on the "Foreman" and the "Students" streams respectively. However, as can be seen in these pictures, the some artefacts resulting from the transmission errors are still noticeable.

The final ER service used in the experiments is the AIR. It is shown that the addition of the AIR service to the other combination of services enables a considerable performance gain in the decoded video quality as observed both in the PSNR measurements and subjective tests. Although the use of the AIR usually enables improved performance, at the instances when the errors are not accumulated (i.e., at the beginning of the video streams) it results in lower performance than the DP, VPR, and HEC enabled service. This is due to the unnecessarily high number of intra-refresh blocks used for the amount of error introduced. However, as the time progresses the advantage of using the AIR service becomes clear with a distinctive performance gain.
Chapter 5. Active Services QoS Support

Figure 5.13: Effects of Error-Resilience Services on the “Foreman” Stream

While the AIR service makes a significant difference on the quality of the decoded “Foreman” stream (i.e., average PSNR result is improved by 2.88dB), the “Students” stream’s average PSNR result is improved by 0.78dB. From this result it can be argued that the video the streams with relatively higher motion activity can obtain a higher benefit than those with lower motion activity.
This is because the DP service cannot be as efficient as it can for lower motion-active sequences and therefore the AIR service can compensate for this deficiency.

The subjective test results illustrated in Figure 5.15 (b) and Figure 5.16 (b) reveal that the best possible results are achievable with the use of the AIR service together with the other services. The use of the AIR service improves the perceptual quality by refreshing the motion-active parts of the frames that are affected from channel errors.
Figure 5.15: Subjective Test Results of the 30th, 60th, 90th and 120th Frames of the Transcoded “Foreman” Stream for a particular seed at $E_b/N_0 = 9$dB: (a) error-free, (b) VPR+HEC+DP+10AIR, (c) VPR+HEC+DP, (d) VPR+HEC, (e) no ER services applied
In summary, the experiments performed showed that the ER services are essential in providing a QoS enabled communications over 3G networks. While it is possible to apply each service individually, the combinational use of all services has been proven to be most effective in preserving the quality of the decoded video. It was also shown that the ER services can show varying performances for different video sequences. That is to say, diverse motion activity levels
amongst different video sequences are the determining factor in the performance of the ER services.

5.3.4 Scene and Channel Adaptive Intra-Refresh

The effectiveness of the ER services, and in particular the role of the AIR algorithm in limiting the temporal error propagation in video communications over W-CDMA networks, was demonstrated in the previous section. As mentioned earlier, the AIR algorithm uses a fixed and predetermined (i.e., before the encoding or transcoding process) number of intra macroblocks for refreshing the frames of a video stream. That is to say, a specific AIR rate is determined before and not altered throughout the encoding/transcoding process. In addition, within the AIR algorithm, there exists no specific mechanism for determining the optimum intra-refresh rate or adapting it to the spatial/temporal characteristics of the video stream with respect to varying channel conditions. Thus, it can be argued that the lack of an adaptation mechanism in the standard AIR algorithm undermines its effectiveness.

In attempt to improve video communications over error-prone channels, various different adaptive intra-refresh mechanisms have been proposed in literature. In [CôtéG] a rate-distortion based method was proposed, which measures the degradation in quality associated with the effects of loosing individual macroblocks, and code certain blocks in intra mode accordingly. Although this was shown to be an effective method, it involves complex computations and can be unsuitable for low latency or real-time video applications. Similarly, [LiaoJ] presented an intra-refresh mechanism that was based on error-sensitivity metrics. This mechanism was implemented at the encoder and modelled the transmission medium as a random error channel with a specific BER. Another intra-refresh approach was employed in [ReyeG] as a tool for providing temporal resilience in a rate-distortion based error-resilience scheme. In this approach, the intra-refresh resilience was altered with respect to the output bit-rate and the BER of the channel. Mean square error measurements were used to calculate the distortion introduced due to the lost macroblocks. Based on mean-square-error measurements, the intra-refresh mechanism alters the number of intra coded blocks in every frame to provide the optimal resilience. This algorithm involves complex computations and can be regarded unsuitable for low latency video communications. A similar approach was also used in [StuhK] where intra-refresh was based on a slightly modified version recommended in H.263 standard. Another adaptive intra-refresh algorithm was proposed in [ChioH], required the encoder to extract some distortion information offline before the transmission, which was later used by a transcoder to decide on the intra-refresh strategy. A more practical intra-refresh method was introduced in [WorrS1]. In this approach, the intra-refresh
mechanism at the encoder was based on a simple activity analysis of each frame and different GPRS channel conditions.

In a networking scenario where the perceived quality of the received video information is characterised by the different requirements of the users and the varying channel conditions, the error-resilience transcoding operation needs to employ simple but effective mechanisms. Such mechanisms are also required to be adaptive to the changing channel conditions and source specific characteristics (e.g., motion activity). In the light of these facts, a novel adaptive intra-refresh algorithm, namely SC-AIR, is proposed in this thesis that is adaptive to varying W-CDMA channel conditions and the activity levels in the video scene. The operation of the algorithm is similar to the standard AIR algorithm, where a motion map is formed to mark the location of the motion-active macroblocks in every frame [MPEG4]. However, the SC-AIR algorithm modifies the number of intra blocks that are coded in the video frame.

While an inadequate number of intra-refresh blocks may be deficient in refreshing the errors in time, excessive number of such blocks may result in increased quantisation distortion especially if the encoding process is rate controlled. The SC-AIR algorithm, which is employed in the error-resilient transcoder, can extract information about the activity levels of a video scene by examining different aspects of the motion-active macroblocks. This information is then coupled with the instantaneous channel condition factor to decide on the optimum number of intra-refresh blocks that is required to obtain the best possible performance in terms of error robustness.

SC-AIR determines the required number of intra-refresh blocks by means of a set of functions, which are used in extracting the scene-activity related information from the input video scene. The outcome of this activity analysis is then modulated with a channel condition function, which represents the signal-to-noise ratio experienced at the downlink W-CDMA channel. The computations of the SC-AIR algorithm are performed for every consecutive predictive frame and do not involve measurable complexity difference compared to the standard AIR algorithm. In other words, the transcoding delay performance of the SC-AIR and the AIR algorithms were measured to be the same. Further analysis and discussion on the transcoder's complexity issues are given in Section 5.3.5.

5.3.4.1 Activity Measurement

In motion compensation based video coders, the sensitivity to error is therefore related with the amount of motion within a scene [WorrS1]. Motion is defined by the motion vectors, and the activity is associated with the motion. Based on these assumptions, it can be claimed that as the amount of activity increases in a video scene, an increased number of intra-refresh blocks may be
required to prevent the temporal error propagation. Thus, activity measurements can be used for developing an adaptation mechanism to counter the adverse effects of changing channel conditions on the perceived video quality.

The activity measurement technique presented here forms the core of the SC-AIR algorithm. It reveals the required number of intra-refresh blocks for every predictive frame. In the development of this algorithm, two different test sequences (i.e., the “Foreman and the “Students”) were used. The unique properties of these video sequences were described in Section 5.3.6.1.

The SC-AIR algorithm is composed of a number of functions which represent the different aspects of the activity in a video scene. The algorithm performs the primary activity analysis using a function named the Normalised Activity Index (NAI). In addition, a number of supplementary functions are also used to shape the output obtained by the NAI analysis. These functions are namely the Motion Macroblock Factor, Range Index, and Scale Factor. The shaped NAI output is used to determine the optimum number of intra-refresh blocks needed for a frame.

Activity Index (AI) is a function computes which the cumulative magnitude of all motion vectors within a frame. NAI is the normalised variation of the AI function with respect to the number of macroblocks within a frame. This normalisation is required so that NAI measurements of different video sequences can be comparable with each other. If the NAI value is high, it is likely that the frame is a part of a highly active scene, which may indicate the need for more frequent intra-refresh (e.g., as in the “Foreman” sequence) than a low motion scene (e.g., as in the Students sequence). The NAI function is can be written as:

\[ NAI(j) = \frac{\sum_{n=1}^{\tau} |mv_f(n)|}{i(j)} \]  

(5.18)

where \(i(j)\) is the number of motion-active macroblocks in frame number \(j\), \(n\) is the macroblock number, \(\tau\) represents the total number of macroblocks in a frame (e.g., 99 for QCIF), and \(mv_f(n)\) is the motion vector (both in x and y directions) of the \(n^{th}\) macroblock in the \(j^{th}\) frame.

The NAI computations for the “Foreman” and “Students” sequences are depicted in Figure 5.17 and Figure 5.18 respectively. As can be seen from this figure, the NAI is able to indicate the activity-level characteristics of the two different sequences.
Nevertheless, the NAI computation on its own is insufficient in terms of defining the accurate levels of activity in a video scene. This is because the output of the NAI function is directly proportional with the motion vector sizes and inversely proportional with the number of motion macroblocks in a frame. As a result, if macroblocks with relatively small motion vectors are
dominant in a particular frame, the NAI parameter will yield a small value, indicating low activity in that scene. This is the case where more frequent refreshing of the scene is required although the NAI result gives low values (e.g., the last 2 seconds of the "Foreman" sequence). Alternatively, if there is relatively small number of motion-active macroblocks but with relatively large motion vectors, then the NAI value will yield a very large value, indicating an extreme activity in the scene. In this case, a high intra-refresh rate will be chosen, which may lead to compromise in the compression efficiency and degradation in the perceptual quality.

The empirical studies performed at the development stage have revealed that the deficiency of the NAI function in determining the accurate levels of activity in a video scene can be compensated using a function called the Motion Macroblock Factor ($\delta$). The product of the NAI and the $\delta$ functions determines the activity. The $\delta$ function is computed every frame and is related with how many motion macroblocks are there in a particular frame. The use of this function can alleviate the anomalies in the NAI decision (as in the last 2 seconds of the "Foreman" sequence). The representation of the $\delta$ function is given in Equation 5.19. It stands for the ratio of the number motion macroblocks to the total number of macroblocks in a frame.

$$\delta(j) = 1 + i(j)$$

(5.19)

Having established its necessity, the next step is to determine the weight of the $\delta(j)$ function. This is the final step in the activity measurement stage of the SC-AIR algorithm, where the product of the NAI and the $\delta(j)$ functions is shaped using other functions like Range Index ($R$) and Scale Factor ($S$). The Scale Factor ($S$) is the weighted representation of $\delta(j)$ and given as:

$$S(j) = \begin{cases} 
0.75 \cdot \delta(j) & \text{if } R(j) \leq 5 \\
\delta(j) & \text{if } R(j) > 5 \text{ and } R(j) \leq 10 \\
2 \cdot \delta(j) & \text{if } R(j) > 10 \text{ and } R(j) \leq 20 \\
3 \cdot \delta(j) & \text{if } R(j) > 20 \text{ and } R(j) \leq 30 \\
4 \cdot \delta(j) & \text{if } R(j) > 30 \text{ and } R(j) \leq 40 \\
5 \cdot \delta(j) & \text{if } R(j) > 40 \text{ and } R(j) \leq 50 \\
7 \cdot \delta(j) & \text{if } R(j) > 50 
\end{cases}$$

(5.20)

and,

$$R(j) = \frac{i(j)}{NAI(j)}$$

(5.21)
where $R(j)$ stands for the ratio between $i(j)$ and the NAI($j$), and is calculated for every predictive frame. $R(j)$ is used to determine the weight of the $\delta(j)$ function. The output of $\delta(j)$ needs to be scaled based on the $R$ function so that a better resolution for the relationship between the NAI and $\delta(j)$ can be reflected on the output SC-AIR algorithm. In other words, $R(j)$ is useful in scaling $\delta(j)$, which shapes the NAI output in a way that an optimum number of refresh blocks are chosen. $R(j)$ defines seven different ranges for the weights of the $\delta(j)$ function. These ranges and the weights of $\delta(j)$ were determined experimentally by observing the effects of the channel errors on the video quality with respect to the number of intra-refresh blocks enabled by the proposed algorithm.

In effect, the ranges defined by the $R(j)$ function correspond to seven different levels of confidence on the NMI decision. As $R(j)$ gets increases, the confidence on the NMI result decreases. Hence, as the $R(j)$ results in higher values, $\delta(j)$ function needs to be scaled with a higher coefficient. Consequently these ranges enable a differentiated refreshing priority between frames with different motion characteristics. For instance, if a frame contains macroblocks with higher average motion vector magnitude than a frame with the same number of motion macroblocks but smaller average motion vector magnitude, then the proposed algorithm will always set a higher refresh rate. In other words, $R(j)$ function maintains a balance between the influence of the number motion-active macroblocks and their average motion vector magnitude in determining the optimum intra-refresh rate.

Figure 5.19 shows the computed $R(j)$ and $S(j)$ functions for the “Foreman” at the transcoder. As can be seen from this figure, the value of the $R(j)$ function falls into its highest range during the last 2 seconds (i.e., between frame numbers 118 and 134) of the Foreman sequence, indicating the need for maximum scaling on the NAI output. Thus, the decision of the $S(j)$ function is scaled accordingly. Similarly, in the region that was indicated as the highest activity region by the NAI function (i.e., between the 9th and the 11th seconds), the output of the $R(j)$ function falls in its lowest range. By this way, over-refreshing and consequently degradation in the compression efficiency is prevented. Other than accounting for these extreme cases, $R(j)$ function is utilised to assign the right weight to the $\delta(j)$ function such that the best possible intra-refresh strategy can be applied.

On the other hand, the low motion nature of the “Students” stream is also reflected on the output of its $R(j)$ function. As depicted in Figure 5.20, the output values of the $R(j)$ function here is far less variable than the case with the “Foreman” stream. Unlike the case with the “Foreman” stream, the $R(j)$ values for this stream are occasionally below 5, which indicates that there are only a few motion-active macroblocks; hence the confidence on the NAI analysis is high.
Having introduced all the functions which play part in the activity analysis of the input video streams to the transcoder, the Intra-Refresh Rate (IRR) required for a given frame can be calculated using the function given in Equation 5.22.

\[ IRR(j) = S(j).NAI(j) \]  \hspace{1cm} (5.22)
The result of the computation of this function for both the "Foreman" and the "Students" streams are shown in Figure 5.21. As can be observed from this figure, the activity measurement functions are effective in differentiating between the activity levels of these two streams and assign particular number of intra-refresh macroblocks for each frame accordingly.

![Figure 5.21: Number of Intra-Refresh Blocks Computed for the "Foreman" and the "Students" Streams](image)

### 5.3.4.2 Channel Factor

For wireless video communications, the varying channel conditions should also be considered in determining the optimum number of intra-refresh blocks to be used. In general, a faster refresh rate is required as the channel conditions worsen, while a limited number of intra-blocks will suffice when the channel conditions become better. The information on the instantaneous downlink $E_b/N_0$ value of the W-CDMA channel is available to the base station. In the proposed algorithm, this information is thus assumed to be fed back to the network node that performs the error resilience services. In 3G systems, such feedback can be provided for the transcoder’s channel adaptation of the algorithm in less than 250ms of delay [TS25.853].

The channel factor $\beta(t)$ is a coefficient, whose values were also determined experimentally for various channel conditions (i.e., $E_b/N_0$) in W-CDMA. $\beta(t)$ is a time dependent function which implies that channel conditions may vary over the time. In experiments conducted, only those
error-patterns corresponding to $E_b/N_o$ ranging from 6dB to 10dB were used. Considering the performance figures presented in [KodiC1], it can be argued that it is not possible to conceive acceptable quality video communications at bit-energy-to-noise ratios below 6dB and the intra-refresh application becomes ineffective. On the contrary, the effects of errors at $E_b/N_o$ rates above 10dB reach to saturation and the channel condition factor can remain the same. The following $\beta(t)$ values were found to provide the optimum efficiency for the SC-AIR algorithm operating at the specified conditions.

$$
\begin{align*}
\beta(t) &= 1.8 \quad \text{if } E_b/N_o = 6dB \\
&= 1.5 \quad \text{if } E_b/N_o = 7dB \\
&= 1.0 \quad \text{if } E_b/N_o = 8dB \\
&= 0.9 \quad \text{if } E_b/N_o = 9dB \\
&= 0.35 \quad \text{if } E_b/N_o = 10dB
\end{align*}
$$

The difference between consecutive $E_b/N_o$ values was chosen to be 1 dB. According to the test results given in [KodiC1], 1dB difference in $E_b/N_o$ value represents a noticeable change in the channel conditions. Hence for the SC-AIR algorithm, any intermediate $E_b/N_o$ figure can be rounded to its closest integer value.

With the addition of the channel factor, the complete SC-AIR algorithm can be formulated as:

$$
\Omega(t, j) = \beta(t).IRR(j)
$$

where $\Omega$ represents the number of macroblocks that need to be refreshed in the motion map of frame $j$, for any channel condition at time $t$.

5.3.4.3 SC-AIR Performance Tests

A series of tests were performed to measure the error-robustness performance of the SC-AIR algorithm in comparison to the standard AIR algorithm. In order to demonstrate the effectiveness of the proposed SC-AIR algorithm, two standard (i.e., the “Foreman” and the “Students” sequences) and two non-standard sequences were used in the experiments. These test sequences were characteristically different form each other in terms of their activity levels and their context. The other reason for choosing these sequences was because they are likely to represent the nature of the typical video services that are likely to be provided in UMTS.

The SC-AIR performance tests involved the execution of previously discussed error-resilience services on the non-resilient video streams. However, the standard AIR service was replaced by the SC-AIR service, and the performance gain obtained by this change was observed. In the
following subsections, firstly, the experimental setup and scenario are explained. Secondly, the experimental results and a discussion about the implications of these results are given.

**Experimental Procedure**

The experimental scenario used in the tests performed for the SC-AIR is similar to the one presented in Section 5.3.3.6. However, the tests performed in the SC-AIR case also involved the use of 2 non-standard sequences (i.e., the “Denmark” and “Interview” sequences). In total, four different sequences were used in order to perform an exhaustive assessment of the SC-AIR performance in terms of applying the best possible intra-refresh strategy for the video streams transmitted over error-prone channels.

Figure 5.22 shows the NAI plots for these sequences as a reference to their activity levels. As indicated from this figure, the “Denmark” sequence is a highly active sequence with large variations in its activity levels. However, the “Interview” sequence is moderately active and in can be classified to stand between the “Students” and the “Foreman” sequences.

![Figure 5.22: NAI Computation for the “Denmark” and the “Interview” Streams](image)

In the experiments performed, two versions of the error-resilient transcoding architecture were used; one equipped with the standard AIR and the other with the SC-AIR algorithm. For all four of the encoded video streams, the same error-resilient transcoding operation (i.e., operation with VPR, HEC, DP, and AIR/SC-AIR services) was applied resulting in 128kbit/s of CBR output.
In order to make a fair comparison, the optimum operation point of the standard AIR algorithm needed to be determined. For this reason, numerous preliminary tests were performed to find the most effective AIR rates. This involved the testing of the AIR algorithm for all four sequences and under various channel conditions (i.e., between $E_b/N_0$ = 6dB and 10dB). As a result, a range of AIR rates were determined experimentally which produced the highest average PSNR values. Depending on the video sequence, the intra-refresh frequency of the AIR algorithm was incremented in steps of 2 to 4 macroblocks to find the optimum operation points. Having determined a set of four possible optimum operation points for the AIR algorithm for every sequence at every $E_b/N_0$ value, the SC-AIR performance was tested with the same video sequences and the channel errors. The results were obtained in the form of PSNR calculations and discussion of these results is given in the following section.

Results and Discussion

After applying ER services with standard AIR or SC-AIR service, the video streams were subjected to various channel conditions. The corrupted streams were then decoded and PSNR calculations were performed. For every channel condition and video stream, four best possible AIR performance figures (i.e., in terms of PSNR results) were compared against the SC-AIR performance. These results are given in Table 5.3, Table 5.4, Table 5.5, and Table 5.6.
### Table 5.3: Test Results for the “Foreman” Stream Using AIR and SC-AIR

<table>
<thead>
<tr>
<th>Eb/No (dB)</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>AIR</td>
<td>24</td>
<td>21.56</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td>21.81</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>32</td>
<td>22.15</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>36</td>
<td>22.11</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>22.37</td>
</tr>
<tr>
<td>7</td>
<td>AIR</td>
<td>24</td>
<td>26.49</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td>26.38</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>32</td>
<td>26.43</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>36</td>
<td>26.54</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>26.81</td>
</tr>
<tr>
<td>8</td>
<td>AIR</td>
<td>12</td>
<td>29.93</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>30.21</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>30.33</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td>30.06</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>30.41</td>
</tr>
<tr>
<td>9</td>
<td>AIR</td>
<td>8</td>
<td>31.01</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>12</td>
<td>31.25</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>31.28</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>31.16</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>31.36</td>
</tr>
<tr>
<td>10</td>
<td>AIR</td>
<td>4</td>
<td>33.04</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>8</td>
<td>33.08</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>12</td>
<td>32.86</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>32.66</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>33.15</td>
</tr>
</tbody>
</table>
### Table 5.4: Test Results for the “Students” Stream Using AIR and SC-AIR

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>avg. 5</td>
<td></td>
</tr>
<tr>
<td>7dB</td>
<td>AIR</td>
<td>3 6 9 12</td>
<td>29.80 30.54 30.47 30.45 30.73</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>avg. 4</td>
<td></td>
</tr>
<tr>
<td>8dB</td>
<td>AIR</td>
<td>3 6 9 12</td>
<td>32.69 32.76 32.62 32.44 33.13</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>avg. 3</td>
<td></td>
</tr>
<tr>
<td>9dB</td>
<td>AIR</td>
<td>3 6 9 12</td>
<td>34.27 34.31 34.23 34.04 34.62</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>avg. 2</td>
<td></td>
</tr>
<tr>
<td>10dB</td>
<td>AIR</td>
<td>2 4 6 8</td>
<td>35.71 34.47 35.39 35.34 35.78</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>avg. 1</td>
<td></td>
</tr>
</tbody>
</table>
### Chapter 5. Active Services QoS Support

#### Interview

<table>
<thead>
<tr>
<th>Eb/No = 6dB</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>22.22</td>
<td>22.21</td>
<td>22.51</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>32</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eb/No = 7dB</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>26.50</td>
<td>26.41</td>
<td>26.38</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eb/No = 8dB</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>8</td>
<td>29.81</td>
<td>29.84</td>
<td>30.14</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eb/No = 9dB</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>8</td>
<td>31.44</td>
<td>31.49</td>
<td>31.23</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Eb/No = 10dB</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>4</td>
<td>33.10</td>
<td>33.08</td>
<td>33.06</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.5: Test Results for the “Interview” Stream Using AIR and SC-AIR
### Table 5.6: Test Results for the “Denmark” Stream Using AIR and SC-AIR

<table>
<thead>
<tr>
<th>Eb/No</th>
<th>Intra-Refresh Algorithm</th>
<th>Intra-Blocks per Frame</th>
<th>Avg. PSNR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AIR</td>
<td>AIR</td>
<td>25.44</td>
</tr>
<tr>
<td>6dB</td>
<td>AIR</td>
<td>AIR</td>
<td>25.50</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>32</td>
<td>25.61</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>36</td>
<td>25.65</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>25.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg. 32</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>28.95</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29.20</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>29.43</td>
</tr>
<tr>
<td>7dB</td>
<td>AIR</td>
<td>AIR</td>
<td>31.96</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>31.97</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td>31.86</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td>31.79</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>32.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg. 19</td>
<td></td>
</tr>
<tr>
<td>8dB</td>
<td>AIR</td>
<td>16</td>
<td>32.70</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>32.85</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td>32.81</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>28</td>
<td>32.69</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>32.91</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg. 18</td>
<td></td>
</tr>
<tr>
<td>9dB</td>
<td>AIR</td>
<td>12</td>
<td>33.99</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>34.03</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>20</td>
<td>33.96</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>24</td>
<td>33.83</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>34.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg. 7</td>
<td></td>
</tr>
<tr>
<td>10dB</td>
<td>AIR</td>
<td>4</td>
<td>33.99</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>8</td>
<td>34.03</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>12</td>
<td>33.96</td>
</tr>
<tr>
<td></td>
<td>AIR</td>
<td>16</td>
<td>33.83</td>
</tr>
<tr>
<td></td>
<td>SC-AIR</td>
<td></td>
<td>34.07</td>
</tr>
<tr>
<td></td>
<td></td>
<td>avg. 7</td>
<td></td>
</tr>
</tbody>
</table>

As can be seen from these results, in all of the experiments performed, the SC-AIR algorithm has resulted in better video quality (i.e., in terms of PSNR) than the AIR algorithm. However, it should be noted that with the channel condition feedback, the SC-AIR algorithm calculates the optimum number of intra-refresh blocks automatically. Conversely, the standard AIR algorithm rates are configured manually to match the performance of the SC-AIR. These results show that with manual configuration of its intra-refresh rates for each video sequence and the channel condition, the standard AIR algorithm’s performance is able to approach that of the SC-AIR algorithm. However, in an actual video communications scenario, such manual configuration will not be feasible and the difference in the performances of these algorithms will hence be much higher.
The major advantage of the SC-AIR algorithm comes from its ability to compute the activity levels in a video scene, taking into account the channel conditions while adjusting its refreshing strategy accordingly. In this way, it can avoid over/under refreshing to take place for a given video sequence under every channel condition. The other advantage of the SC-AIR algorithm is that it operates in real-time without causing a considerable delay overhead. The details about the imposed complexity of this algorithm on the transcoding operation are given in Section 5.3.5.

5.3.5 Transcoder Complexity

Other than providing various services for networked video applications, the most significant design requirement from a transcoder is the computational efficiency. In other words, the transcoding operation should be able to compute in real-time such that the delay introduced by its services does not impede the application performance. In the previous sections, the performance analysis of the transcoder services in terms of providing robust video communications over the 3G network have been presented. In this section, methods which have been used to simplify the complexity of the transcoding architecture are presented. In addition, the complexity analysis of the error-resilient video transcoder is performed to assess its delay performance and determine its suitability for implementation in the network nodes.

The CPDT architectures are inherently complex but offer the flexibility of implementing variety of services (e.g., AIR, SC-AIR, and Rate-Control). Thus, complexity reduction techniques need to be employed to enable their practical feasibility. These simplifications are usefully performed on the DCT and motion compensation operations. The simplifications in the DCT and inverse-DCT operations involve reducing the number of iterations for every macroblock by using a marker which signs the end of non-zero coefficients [YounJ]. For the motion compensation operation, complexity reduction can be realised by removing the DCT block and using the same motion compensation block with the decoder. However, this can introduce the drift-effect [VetrA], [XinJ], which causes severe degradation to the video quality. Although, various techniques have been developed to tackle the drift-effect [AssuP], [BjorN], [YounJ], cascaded pixel-domain transcoding architecture was chosen to avoid the problem. This transcoding architecture uses two separate motion compensation units for both the decoder and encoder sides. However, this results in complexity due to the inverse-DCT block and the frame buffer required for the encoder side motion compensation process.

In order to compensate for the increased complexity, DCT/inverse-DCT operations were optimised using the image processing libraries for DCT transformations, which were designed by Intel® Integrated Performance Primitives [Intel]. These libraries make optimised use of the
processor architecture and reduce the computation time. The PC used for the development of the transcoder software was equipped with an Intel® Pentium® 4 processor. Thus, the processing speed gain obtained by using these image processing libraries was around 30%. In other words, the transcoding speed was increased by 30%.

Further simplifications for the transcoder can be achieved by reusing some of the information available from its decoding operation. For instance, it is known that the motion estimation process of a standard encoder constitutes almost 60%-70% of the total encoding complexity [XinJ]. However, it is not necessary for a transcoder to perform motion estimation. Instead, the motion information which is already available after the transcoder’s decoding process can be utilised. If preferred, motion refinement techniques can be applied on the existing motion vectors such that they become optimised for the rate-controlled output video stream. Although this approach performs better than a simple motion reuse scheme, it involves a certain level of complexity that is directly proportional with the refinement window size. Therefore, based on the motion refinement performance figures given in [SadkA], it was decided that motion reuse scheme is more suitable for this transcoder when the trade-off between the operational-performance and perceptual video quality is concerned.

In addition to the motion vector reuse scheme and the fast DCT operations, three other optimisations were made during the development of the transcoder. Amongst these, reuse of some of the header information, omission of unnecessary initialisation routines at the encoding section, and when appropriate, reuse of the same allocated memory and functions for both encoder and decoder computations can be listed.

Having implemented the aforementioned optimisations the transcoder’s complexity analysis was performed. This involved testing the computational complexity of the transcoder software under the experimental scenario given in Section 3.2. The transcoder software was tested on a PC with an Intel® Pentium® 4 processor at 3.19GHz, with 512kbytes of L2 cache, 1GByte of memory running the Windows XP operating system. The operation speed of the transcoder is represented with the number of frames transcoded per second (fps). The computational complexity analysis of the transcoder for the “Foreman” and the “Students” sequences are given in Table 5.7 and Table 5.8 respectively. Initially, the transcoder was first operated with no services turned on. In such case, the transcoder’s operation only involves decoding and re-encoding with the same motion vectors and with the quantisation parameter used at the encoder. Then, the RC and ER services were progressively turned on to examine their effects on the operation speed.
The obtained results reveal that the DP service is the most complex of all other services while VPR and HEC services constitute negligible complexity. These figures also show that the SC-AIR algorithm is only as complex as the AIR algorithm. The transcoding speed is also dependent on the bit-rate of the incoming bit-stream. In other words, the decoding complexity increases with the increasing bit-rate of the input bit-stream, which in turn is manifested as reduction in the transcoder’s operational speed. As the bit-rate of the incoming video stream increases, there will be more texture and motion information to decode, hence the transcoding speed will be decreased. Rate-control is also a determining factor on the operational speed. Naturally, as the output bit-rate reduces, there would be less information to encode, resulting in a faster transcoder operation as displayed with the performance figures associated with different RC rates.

<table>
<thead>
<tr>
<th>VBR Foreman @avg. 445kbits/s</th>
<th>ER Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Service</td>
<td>VPR+HEC</td>
</tr>
<tr>
<td>No RC</td>
<td>178 fps</td>
</tr>
<tr>
<td>RC@256kbits/s</td>
<td>190 fps</td>
</tr>
<tr>
<td>RC@128kbits/s</td>
<td>204 fps</td>
</tr>
<tr>
<td>RC@64kbits/s</td>
<td>209 fps</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>VBR Students @avg. 230kbits/s</th>
<th>ER Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Service</td>
<td>VPR+HEC</td>
</tr>
<tr>
<td>No RC</td>
<td>235 fps</td>
</tr>
<tr>
<td>RC@128kbits/s</td>
<td>232 fps</td>
</tr>
<tr>
<td>RC@64kbits/s</td>
<td>238 fps</td>
</tr>
<tr>
<td>RC@32kbits/s</td>
<td>245 fps</td>
</tr>
</tbody>
</table>

The operation speed of the transcoder can also be represented as the transcoding delay by calculating the time required to transcode a single frame. Based on the maximum and minimum values given in Table 5.7 and Table 5.8, the amount of delay introduced to the end-to-end delay budget by the operation of the transcoder varies between 4.1-6.5 ms.

It should be realised that these figures are dependent on various video coding parameters and can vary significantly based on the given computational resources in the networks. Nevertheless,
based on the given operation scenario, these performance figures show that the transcoding operation is practically feasible in real-time and therefore can be implemented in the active nodes.

5.4 Active Congestion Control

In earlier sections, the benefits of utilising the transcoding class services over the UMTS network have been demonstrated. However, uses of the proposed transcoding architecture are not limited to the wireless networks. The rate-control and error-resilience services can also be utilised for the fixed networks. This would alleviate the congestion in the network and improve the perceptual quality for the video applications.

Building up on the DiffServ concept introduced in the previous chapter, this section is aimed at introducing the possible advantages of active services support for multimedia communications over the fixed networks. For this purpose, an experiment scenario was devised where transcoding class services provide active congestion control for video streams that are transported using the AF mechanism of the DiffServ network. By utilising the capabilities of the transcoding class services, possible network congestion can be tackled when and where necessary without any means of support from the network end-points.

To demonstrate the use of transcoding class services for congestion control, an experimental scenario similar to the one introduced in Section 4.8.1 was used. However, this time no network bottleneck was implemented. This is because the aim of this section is to present an alternative method for video traffic conditioning at the network edges. Traditionally, traffic conditioning involves token bucket and algorithmic dropping mechanisms as introduced in Section 4.4.1. However, it is argued here that such mechanisms may not be suitable for multimedia applications unless applications are specifically designed for supporting graceful quality degradation (as explained in Section 4.8.3).

Therefore, the proposed transcoding architecture is installed on a network node to provide the necessary traffic conditioning functions for a video stream in an AF serviced traffic aggregate. In other words, the transcoder is responsible for regulating the input video traffic streams' bit-rate such that they abide by their TCSs as well as making better utilisation of network bandwidth. However, unlike the traditional approaches, it is advocated here that the admission to network resources should not be limited to a fixed data rate. That is to say, if a network client purchases a particular service with a specific rate, he/she should not be bound to transmitting/receiving at that particular rate only. If the network resources for a particular AF class are underutilised, then applications should be allowed to consume more bandwidth than their standard service rates. For
example, standard transmission rate defined in the TCS can be 128kbits/s but if the network is capable of providing more bandwidth at any given time, applications should be able to increase their transmission rate to 256kbits/s. However, if the network is likely to become congested, the users should also be ready to accept lower rates (e.g., 64kbits/s). As a result, applications can avoid packet losses and uncontrolled quality degradation in the application quality.

The function of a transcoder at the edge of a DiffServ network is to monitor the levels of incoming traffic and determine how video applications should be conditioned to adapt to the changing congestion levels. The logical structure of the traffic conditioning mechanism using the transcoding class services is shown in Figure 5.23.

The input traffic aggregate is first classified into their corresponding PHBs. The classification process also includes the identification of different traffic streams within the aggregates of each PHB. In other words, packets with lower and higher priorities are differentiated. Higher priority packets are subject to traffic conditioning operations so that their traffic volumes do not exceed the specified limits (e.g., as specified in their TCS). Nevertheless, higher priority traffic streams rely heavily on the video transcoding mechanism such that lower priority streams are effectively conditioned (i.e., transcoded) and consequently packet losses are minimised.

The focus of the experiments was narrowed down to a single PHB to investigate the advantages of the transcoding operations. The assumption made here is that the operations performed on each PHB are independent of the others. Therefore, it was not necessary to investigate all PHBs given in the scenario described in Section 4.8, where the network traffic is treated using differential resource allocation and scheduling amongst a number of different service classes.
Chapter 5. Active Services QoS Support

5.4.1 Experimental Procedure

To demonstrate the advantages of employing transcoding class services in the fixed packet networks two different experiments were devised. The first experiment demonstrated the case where no transcoding was employed and the video application’s bit-rate was kept constant. In the second experiment, transcoding operations were introduced, and effects of dynamically regulating the video bit-rate on the video performance were investigated.

In the experiments, two different kind of traffic sources were used to compose the traffic aggregate, which was classified to a single PHB. The first source was a CBR video stream, and the other was a collection of VBR sources which represented a mixture of TCP and UDP traffic. The VBR traffic was modelled using the Pareto distribution as described in Section 4.8.1.1. A number of traffic sources with different start and stop times were used to simulate the changes in the network traffic load. Each source had a throughput of 64kbits/s on average, and was configured to produce 576bytes of packets with varying inter-arrival times. The CBR video traffic source, on the other hand, was designed to produce 10fps and QCIF resolution MPEG-4 traffic at constant bit-rate of 128kbits/s for the first experiment, and 256kbits/s for the second. The GOP structure for the video stream was chosen as (19, 1), representing the use of an I-frame at every 2 seconds. The video frames were packetised into 576byte packets. The video source represented the "Interview" video sequence. This sequence was chosen since it is long enough to demonstrate the effects changing the transcoding rate multiple times during the simulation. The total simulation time was set to 40 seconds, which corresponds to the transmission of 400 video frames for a video stream encoded at 10frames/s frame-rate.

In the first experiment, all packets were assumed to have the same priority. Therefore, in the case of congestion, packets were dropped indiscriminately. In the second experiment, the CBR video stream was assumed to have a lower priority than the VBR traffic; hence it bore the responsibility of regulating its bit-rate during the congestion times. That is to say, in the case of congestion, VBR traffic should have lower drop precedence. In the classical DiffServ scenario, traffic streams with higher drop precedence are more likely to be shaped/policed (e.g., by dropping packets) than the traffic streams with lower drop precedence. However, the objective of utilising transcoding class services is to minimise, if not eliminate, the possibility of packet drops by dynamically adjusting the bit-rate of video traffic. As a result, over-utilisation of the bandwidth would not take place and the packet drops would be minimised.

It should be noted here that in a real life scenario such services may not be limited to servicing lower priority traffic streams only. That is to say, based on the agreements between the service
provider and the network clients, transcoding operations can be applied to high, low, and lower (i.e., all three drop precedence in an AF service) priority video streams of an aggregate.

For convenience, from this point onwards, the CBR video stream is referred as “video-stream” and the VBR traffic streams are referred as “background-stream”. The queue capacity for their aggregate was determined as 50kbits, and the service rate was set to 1Mbits/s (representing the total bandwidth made available to this traffic aggregate). The fundamental assumption for the second experiment was that the agreement of the video stream user with the service provider specifies that the standard transmission rate is 128kbits/s. However, up to 256kbits/s can be supported if the network conditions allow. Yet, 64kbits/s is the lowest and the promised service level that will be supported if the network conditions deteriorate. For this purpose, the transcoding service is initiated at the network edge in order to provide dynamic adaptation to the varying availability of the network resources.

The video-stream source transmitted at 128kbits/s for the first experiment, and 256kbits/s for the other. For both experiments, the background-stream started transmitting at around 512kbits/s, which leaves 489kbits/s for the video stream out of the total 1Mbits/s bandwidth. Later, the number of active traffic sources creating background stream was gradually increased. Towards the middle of the simulation, the total bandwidth consumed by the background stream reached to average 872kbits/s.

In the second experiment, the dynamic bit-rate adaptation operation performed by the transcoder needed to be aware of the bit-rate of the incoming background-stream. This information was provided from the traffic measurements made by the meter block, and is represented in terms of average bits-per-second (bits/s). As a result, the transcoder can only react to a change with 1 second delay. If the traffic level exceeds a certain threshold, the transcoding operation takes place to prevent any packet loss (i.e., video bit-rate is reduced) or to make better utilisation of the network resources (i.e., video bit-rate is increased). Based on the available bandwidth for the video-stream, the transcoder was configured to follow a 3-step rule. That is:

\[
\text{If} \ (\text{Background-Stream} \leq 64\% \text{ of Total Bandwidth}) \\
\quad \text{Do not Transcode} \\
\text{If} \ (64\% \text{ of Total Bandwidth} < \text{Background-Stream} \leq 77\% \text{ of Total Bandwidth}) \\
\quad \text{Transcode @ 128kbits/s} \\
\text{If} \ (\text{Background-Stream} > 77\% \text{ of Total Bandwidth}) \\
\quad \text{Transcode @ 64kbits/s}
\]
Chapter 5. Active Services QoS Support

When the video-stream is downscaled to lower bit-rates, the transcoder is set to apply all error-resilience services except the SC-AIR service (i.e., AIR is used with the refresh rate of 8). In doing this, it is intended to provide maximum protection against the effects of any packet loss.

Both experiments involved collecting statistics about the packet losses experienced and measuring their effects on the video performance by using both subjective and objective tests.

5.4.2 Results and Discussion

Two different experiments were performed to test the performance of introducing transcoding class services into the network. In the first experiment, the video-stream source was set to produce CBR traffic at 128kbits/s (as specified in the TCS) from the beginning of the simulation until the end. At the same time, the background-stream gradually increased its transmission rate and caused congestion from the 10th to 25th seconds. The traffic pattern of the background-stream is given in Figure 5.24. The markings on this graph depict the congestion period. During the congestion period, the background-stream is likely to exceed its maximum allowed bandwidth (872kbits/s), and on average it was measured as 880kbits/s. However, due the bursty nature of this traffic, the bandwidth consumption can momentarily increase up to 1Mbits/s. The fact that UDP traffic cannot respond to congestion means that the congestion can last for an extended period of time. However, it is assumed here that the background-traffic is a mixture of UDP and TCP traffic and the congestion is alleviated after a period of time (i.e., after the 25th second).

The traffic conditioning of the traffic aggregate was performed with an absolute dropper. This is realised with a FIFO queue with 50kbits depth and 1Mbits/s service rate. Therefore, if the total queue size exceeded 50kbits, then the incoming packets would be dropped. Therefore, in the first experiment, both background-stream and video-stream experienced packet loss when the aggregate bit-rate exceeded 1Mbits/s. As described in the scenario explained previously, the background-stream is bursty, which means that it can occasionally exceed its intended bandwidth. The result of the first experiment has shown that unless compensatory measures are taken (e.g., transcoding), the packet loss experienced during the congestion period can reach up to 3% for the background-stream and 15% for the video-stream. The difference in packet loss rates is due to the inter-arrival time differences between two traffic types. A similar result was also demonstrated in Section 4.8.3.

In the second experiment, the employment of the transcoder had a significant quality on the video application. The rate-control algorithm of the transcoder provided adaptation to the bandwidth fluctuations and prevented packet losses in most of the case. The experiment was repeated for 15...
times with different seeds. Although no packet loss was detectable in most of the simulation runs, rare packet loss events were also observed for both video-stream and background-stream. The maximum packet loss rate observed was around 0.7% for the video-stream, and 0.14% for the background-stream. However, the error-resilience services used in the transcoding process managed to overcome the any adversity caused by such losses.

![Figure 5.24: Traffic Pattern of the Background-Stream](image)

PSNR measurements were performed on the decoded video-stream for both experiments. Figure 5.25 shows the results of these measurements.

![Figure 5.25: PSNR Plots of With and Without Transcoding Cases for the “Interview” Stream](image)
The dotted brown line shows the PSNR plots of the 128kbits/s CBR “Interview” stream when there is no packet loss. In this sense, it is the reference line for the other two lines. The red line is the result of the first experiment and shows how the quality of this video is degraded during the congestion period. During the congestion, the measured average PSNR value is 25dB. However, the fact that the PSNR measurements taken between frame number 134-180 and 200-240 indicates that the video quality is severely degraded and the visual content can be regarded unusable. A momentary recovery between frame numbers 182 and 193 is due to the decoding of an uncorrupted I-frame. After the congestion period, the video quality is recovered by the I-frames used in every 2 seconds, and no degradation in the video quality can be observed. It should be noted here that the simulation result used to demonstrate the outcome of the first experiment was obtained from one of the simulation runs which only resulted in 11% packet loss for the video-stream during the congestion period. However, it was observed that this figure could increase to 15% in other runs.

The blue line represents the outcome of the second experiment. The worst simulation result that included a packet loss rate of 0.7% during the congestion period was chosen for PSNR measurements. The intention in doing that was to demonstrate that even in the worst case, the transcoder is able to salvage the video quality by dynamically adapting the video bit-rate. At the beginning of the simulation, the transcoder measured the bit-rate of the incoming background stream and decided that the bandwidth is underutilised. As a result, it chose not to transcode the video-stream. For this reason, the video was transmitted at 256kbits/s for the first six seconds of the simulation, which resulted in 3.4dB better performance than the case in the first experiment\(^1\). As the utilisation of the bandwidth increases, the transcoder decides to regulate the incoming video traffic, and hence reduces the bit-rate to 128kbits/s after the 6\(^{th}\) second. From this point until further change in the bit-rate, the PSNR measurements taken matched with those taken in the first experiment in the same time interval. On the 11\(^{th}\) second, a possible congestion is detected and the video bit-rate is throttled back to 64kbits/s. As can be seen in Figure 5.25, during the congestion period the average PSNR value was measured as 30dB. This has resulted in an average of 5dB gain in comparison to the results of the first experiment. After the 25\(^{th}\) second, the congestion was alleviated. Thus, from the 26\(^{th}\) second onwards, the transcoder gradually increased the bit-rate of the video stream. After the 36\(^{th}\) second, the transcoder was able allow 256kbit/s for the video-stream, and as a result PSNR measurements have improved by 2.9dB on average.

Subjective tests were also performed to assess the video quality in both experiments. An illustration of the results is given in Figure 5.26.

\(^1\) Every second corresponds to 10 video frames.
Figure 5.26: Subjective Test Results of the Experiments using the “Interview” Sequence: (a) 2\textsuperscript{nd}, 36\textsuperscript{th}, 60\textsuperscript{th}, and 90\textsuperscript{th} frames in the first experiment, (b) 139\textsuperscript{th}, 159\textsuperscript{th}, 179\textsuperscript{th}, and 228\textsuperscript{th} frames in the first experiment (congestion period), (c) 286\textsuperscript{th}, 327\textsuperscript{th}, 366\textsuperscript{th}, and 398\textsuperscript{th} frames in the first experiment, (d) 2\textsuperscript{nd}, 36\textsuperscript{th}, 60\textsuperscript{th}, and 90\textsuperscript{th} frames in the second experiment, (e) 139\textsuperscript{th}, 159\textsuperscript{th}, 179\textsuperscript{th}, and 228\textsuperscript{th} frames in the second experiment (congestion period), (f) 286\textsuperscript{th}, 327\textsuperscript{th}, 366\textsuperscript{th}, and 398\textsuperscript{th} frames in the second experiment.
The subjective test results also show the significant quality gain obtained by introducing dynamic bit-rate adaptation in the network. The visual quality enhancement effects of the developed transcoding architecture are particularly perceptible at the congestion period as illustrated in Figure 5.26(b) and (e).

In summary the advantages of employing transcoding class services in the fixed networks would bring in a number of important advantages. These are:

- Prevention of packet loss for both video and other applications.
- Improvement of video quality through better utilisation of the network bandwidth.
- Salvaging of the video quality during the congestion times.

Overall, providing transcoding services in a network with active networking capabilities presents an efficient and alternative method for traditional traffic conditioning mechanisms. It should also be noted here that, although the simulations performed have assumed a DiffServ scenario, it would be equally advantageous to employ similar active services in packet networks providing only best-effort treatment for its traffic.

5.5 Concluding Remarks and Recommendations

This chapter has presented a user-centric active networking service, namely the transcoding class services. Transcoding class services provide customisation of the video streams based on the requirements dictated by the applications, users, and the service providers. The focus of the research presented in this thesis was on the development of advanced services that can utilise the functionalities provided by the active networking platforms. Therefore, an active networking platform was assumed to be available, and the author relies on respective efforts in the development of a functional and well performing active networking platform.

The transcoding class services are composed of two different types of transcoding services. These are rate-control (RC) and error-resilience (ER) services. The services were developed on a low-complexity CPDT architecture that was developed as a part of the research work presented in this thesis. The RC service is the transcoding operation that is used to smooth out the fluctuations in the throughput and convert input high bit-rate streams into lower rates. The objective of this operation is to reduce the bit-rate while maintaining low complexity and achieving the highest quality possible. RC service is particularly useful for tailoring the input video streams to the capabilities of different user-terminals (i.e., different users may have different QoS agreements.
with their service provider). A slightly modified version of the TM5 algorithm was used to achieve the required rate-control functionality.

Exhaustive computer simulations were performed to assess the effects of introducing RC services on the perceived video quality. The experiments have shown that the use of rate-control has a significant effect on the decoded video quality. It was found that the perceptual quality attainable in any frame is inversely proportional to the total number of bits required to encode the motion information of the same frame. Moreover, differences in the level of residual texture information between the frames mean that each frame is subject to different level of compression. Therefore, applying the RC service on the input VBR streams causes fluctuations in the perceived video quality. In TM5, such fluctuations become perceptually more noticeable for video sequences with changing motion-activity levels and/or scene changes. Nevertheless, the RC service is effective in maintaining a constant output bit-rate for the transcoded streams.

On the other hand, ER services are used to provide protection for the video streams prior to their transmission over a noisy channel or a congested network link. Noisy channel conditions and packet losses due to congestion introduce errors in the compressed video streams, which are manifested as degradations in the perceived quality of the decoded video. ER services include error-resilience tools such as VPR, HEC, DP, AIR, and SC-AIR. The proposed SC-AIR algorithm is a scene and channel adaptive intra refresh scheme which analyses the video content to measure the scene activity and uses the channel condition feedback to determine the optimum intra refresh frequency for every video frame.

The performance of the ER services was tested over the W-CDMA channel. The results have revealed that the use of these services are absolutely essential and can provide significant improvements on the perceived video quality (e.g., around 5-8dB PSNR difference compared to the error-prone case). Through exhaustive experimentations, it was also demonstrated that for any given channel condition, SC-AIR can yield superior performance compared to using constant frequency AIR algorithm.

The proposed Active Congestion Control (ACC) scheme also utilised the transcoding class services as an alternative traffic conditioning mechanism for DiffServ networks. The transcoder was assumed to be operational on an active edge node for regulating the input video traffic streams' bit-rate such that they abide by their TCSs as well as making better utilisation of network bandwidth. Computer simulations were performed to illustrate the advantages of the ACC scheme. The results have shown that ACC can effectively prevent packet-losses and improve the perceived video quality both during the congestion and non-congestion periods. In summary,
ACC can provide prevention of packet loss for both video and other applications, improvement of video quality through better utilisation of the network bandwidth, and salvaging of the video quality during the congestion times.

In order not to impede the application performance, transcoding operation should be able to compute in real-time. A number of measures were taken during the design of the transcoder such that its operation is simplified. These were, motion vector reuse, optimised DCT operations, reuse of some of the header information, omission of unnecessary initialisation routines at the encoding section, and when appropriate, and reuse of the same allocated memory and functions for both encoder and decoder computations. Furthermore, complexity analysis of the error-resilient video transcoder was performed to assess its delay performance and determine its suitability for implementation in the active network nodes. The results have shown that the software implementation of the transcoder on a desktop pc using Intel® Pentium® 4 processor at 3.19GHz, with 512kbytes of L2 cache, 1GByte of memory, and running the Windows XP operating system results in around 5ms of transcoding delay. This proves that the transcoder can be operated in processing capable network nodes without hampering the real-time operation of the video-based applications.

In summary, transcoding class services can provide support for various QoS related problems which are due to the heterogeneities in network architectures, and end-user capabilities/requirements. However, the ultimate QoS support can only be provided when all class of different active services are made ubiquitously available in the networks.
Chapter 6

6 Conclusion

6.1 Preamble

The quality of service (QoS) provided in traditional packet-switched networks is limited to efficient routing protocols and congestion control techniques. This approach, although being suitable for data transfer, is barely effective for real-time and low-latency multimedia applications due to the limitations imposed by local bandwidth scarcity, channel variations in wireless access networks and hardware restrictions of user terminals. In addition, the support for multimedia applications is defined by standard protocols, which limit the flexibility and scalability required for multimedia applications. Therefore, an alternative networking architecture is required to account for these limitations and provide the end-to-end support for enabling maximum possible QoS provision. It is envisaged that such architecture will be based on active network notion. Deploying active routers within the delivery network can alleviate the bandwidth and channel restrictions by executing active services, i.e. adaptive transcoding algorithms. This also enables user-specific computations to take place inside the network, which can tailor the multimedia applications to the specific needs of the users.

6.2 Concluding Overview

In the view of the research work presented in this thesis, multimedia applications can be provided with enhanced QoS support through the use of active networking technologies. The aim of this research was to enhance the QoS performance of the real-time/low-latency video applications by using a series of active services. The active services developed in this research are video transcoding based solutions which can alleviate problems related with network heterogeneity, varying channel conditions, bandwidth fluctuations, and user requirements. A number of steps have been followed in developing the active services approach. Firstly, the requirements of users, applications, and networks have been studied. At the same time, the functionalities of existing IP QoS solutions have been investigated. Certain problems associated with such solutions have been highlighted and the related research activities have been introduced. The preliminary investigations performed have revealed that service differentiation architecture has significant
advantages in providing QoS for the multimedia applications in bandwidth limited networks. However, it is also necessary to complement and enhance service differentiation and best-effort architectures with a dynamic content adaptation interface. This interface is the active networking technologies and its services are the active services. The different investigation phases carried out as a part of this research work are organised in separate chapters. In this section, the concluding overview of these chapters is provided.

Chapter 2 comprises the background for QoS provision in multimedia communications. It introduces the understanding of QoS and elaborates this concept form the network, application, and user point-of-views. The networking point of view is usually adopted by the service providers and is used to define the QoS offered to the network clients. According to this view, delay, jitter, packet loss, loss pattern, and the bandwidth are the parameters that determine the QoS received by the applications. On the other hand, application point-of-view defines specific requirements regarding to these parameters. Each parameter has a different effect on the application performance. From the user point of view, the parametric measurement of application performance cannot be directly mapped onto the perception of the media quality. In general, the user perception of quality is affected by a number of psychological factors including application response time, predictability, expectancy, and consistency. Therefore, in order to design an effective QoS architecture, all such different requirements should collectively be considered. In the research work presented here, the main focus was on optimisation of the application performance with respect to the network impairments.

The IP protocol is seen as the common interface that enables interoperability between different networks. Therefore, this chapter has also provided an insight to the existing IP QoS approaches in both wired and wireless networks. It can be claimed that due to massive deployment and adoption of IP based communication infrastructures, it is unlikely in near future to witness the emergence of an alternative networking protocol that would change the fundamentals of the way networks operate today. Therefore, the efforts on building an effective QoS framework should be within the scope of IP based communications. However, present-day IP QoS architectures are not able to provide extensive QoS support for multimedia applications due to the heterogeneities existing at various levels of the communication systems. The solution to such problems have conventionally been realised by employing QoS adaptation mechanisms that can be introduced at the end-points and intermediate network nodes. A number of adaptation mechanisms have been discussed with a particular emphasis on network-based mechanisms such as transcoding proxies/gateways. However, such approaches tend to form short-term solutions for the QoS problems, and they lack in scalability and flexibility. To provide a comprehensive and long-term solution, networks are required to evolve form their closed system form into an open,
programmable, and adaptable form that can suitably adapt the applications to the requirements of the users as well as the networks. In effect, networks should move from the concept of being a data delivery medium to a service delivery medium. In this thesis, it is envisaged that this evolution will be through the path of active networking concept.

Chapter 3 presents an introduction to active networks which is a new and radical concept in packet communications. Active networking describes the ways in which present-day networks are transformed into dynamically programmable and fully flexible mechanisms that can offer intelligent services both at the control and data planes. The active networks concept advocates the need for having open and programmable network devices that can perform customised computations on the user packets and dynamically deploy new protocols without having the need for standardisation. In essence, it networks become virtual extension of the user terminals and provide the required media customisations/adaptations on behalf of the user. From this point-of-view, it is claimed that active networking is contradicting with the end-to-end arguments, which are the basic principles that have enabled the success of the Internet. However, with the changing face of the networks, applications, and user expectations/requirements, the end-to-end arguments should be carefully reviewed and not applied in a definitive way. It is in the view of the research presented in this thesis that it is scientifically reasonable to question the applicability of end-to-end arguments and seek for alternative approaches for future networking systems. In addition, the divergence from this view is already being seen in today’s networks where additional intelligence and processing is implemented in the networks by the IP QoS architectures (i.e., DiffServ).

The applications of the active networks can be categorised into three areas; QoS, protocol deployment and signalling, network management. In this thesis, only the QoS related research have been investigated. The QoS related applications of active networks are likely to motivate service providers and vendors to adopt active networking technologies. This is because, active networks have the potential to create new and better ways of deploying services in the networks and charging their customers without the need of long and cumbersome standardisation processes. The users will benefit from the increased service flexibility which can allow them to choose and customise the services they demand, and pay according to the exact QoS they receive. In addition, third party service developers will be introduced in the market, which will create competition amongst different service providers and result in better services at lower prices for the users.

Deployment of active networking components in the networks is not far from realisation; thanks to the rapid advances in the semiconductors technology that will enable extensive processing and memory resources to be available in the networks at affordable costs. Nevertheless, it is commonly accepted in the research community that the network core should be kept as
simple/fast as possible. Therefore, the network edges will have to take the extra processing burden essential for active processing.

Currently, the active networking research is at its experimental stage and there exists no single active networking model with a generic interface that can be accessed and programmed by various types of user programming models. Therefore in research work presented in this thesis, no particular active networking approach has been elected as the base model. The main purpose of studying active networks concept was to explore the possibilities of improving multimedia QoS with the use of AP inside the networks. Therefore, the focus is put on the development of active networking services which are not complex and hence can be implemented in the active nodes.

Chapter 4 introduces the DiffServ architecture which enables service differentiation for different categories of end user applications. In the view of the research work presented here, the active networking technologies can make significant contribution towards complementing the existing IP QoS service architectures. Unless active networking technologies are implemented ubiquitously in networks, they cannot offer effective solutions in providing end-to-end support for user applications. Instead, active networking components are likely to be deployed progressively, and work in cooperation with the existing architectures. In addition, adoption of such strategy can prove to be more cost effective and speed up the deployment of the DiffServ architecture.

The DiffServ model is based on relative priority treatment of different classes of traffic. The motivation behind the development of this model is the stringent requirements of the real-time multimedia applications and the need for service differentiation due to competitive nature of the marketplace. It recognises and categorises the traffic streams of different applications into behaviour aggregates based on their common QoS requirements. Then, each traffic aggregate is given a predetermined level of QoS within the DiffServ domain. A DiffServ domain is characterised by a cloud model with edge routers at the borders and the core routers in the centre. Edge routers are responsible for performing complex classification and conditioning operations, while the core routers, due to the large volumes of traffic they carry, are only expected to forward the incoming traffic towards their destination. The forwarding procedure involves differential and policy based traffic conditioning, queueing, and complex scheduling operations, which isolate the QoS received by a particular traffic aggregate from the others.

To demonstrate the possible advantages of utilising the DiffServ QoS architecture, a simple model was constructed and series of experiments were performed. This has also served as the basis for further analysis performed with the introduction of active networking services. It was shown by the experiments that the best-effort treatment of multimedia streams cannot provide a satisfactory
QoS performance for the multimedia applications, and the applications are subjected to uncontrolled packet losses in the case of network congestion. Such losses have severe effects on the perceived media quality. In the case of video-based applications, it was shown that there is an optimal average bit-rate, which is somewhat independent of the packet loss rate but more dependent on the spatio-temporal characteristics of video sequence. This indicates the necessity of employing congestion adaptation mechanisms for such applications. When the DiffServ model was introduced in the network, it was observed that the QoS given to different traffic aggregates became independent of each other’s traffic volume. This was achieved at the expense of a slight increment in the end-to-end delay performance. Thus, unless the traffic streams of the same aggregate do not exceed their predetermined limits, it is unlikely to experience significant packet loss in a well managed DiffServ network. However, depending on the granularity of traffic control applied on individual aggregates, transient packet losses can occur, which may have adverse effects on the application performance. It is argued in this thesis that rather than expecting the users to tolerate such effects, applications should be adapted to the fluctuations in the network bandwidth to minimise the possibility of any packet loss.

Finally, in Chapter 5 transcoding class services has been introduced which provide customisation of the video streams based on the requirements dictated by the applications, users, and the service providers. These services are assumed to be operational in network nodes with active networking capabilities. The transcoding class services are used for providing rate-control and error-resilience for the MPEG-4 video-based applications. These services are performed on a CPDT architecture that was developed as a part of the research work presented in this thesis. The rate-control is used to smooth out the fluctuations in the application bit-rate (conversion form VBR to CBR) and scales the input stream down to lower bit-rates. This service is useful for tailoring the input video streams to the capabilities of different user-terminals (i.e., different users may have different QoS agreements with their service provider), and providing link adaptation in the case of network congestion. The transcoding operation causes degradations in the video quality due to the requantisation process. However, experiments with UMTS and DiffServ networks scenarios have shown that the potential advantages of such operation clearly outweigh its disadvantages.

The error-resilience services are useful in providing protection for the video streams prior to their transmission over noisy channels or congested network links. Such channel conditions introduce errors in the compressed video streams, which are manifested as degradations in the perceived quality of the decoded video. The performance of the error-resilience services was tested over the W-CDMA channel. The results have shown that the optimum performance is attainable when all error-resilience tools (i.e., VPR, HEC, DP, and AIR) are used together. Compared the error-prone case, the performance gain obtained (e.g., in terms of PSNR), depending on video sequence, was
between 5 to 8dB. In addition, a unique intra-refresh algorithm, namely SC-AIR, was developed which optimises the number of intra blocks in a video frame based on the scene activity and the channel condition. Through exhaustive experimentations, it was demonstrated that for any given channel condition, SC-AIR can yield superior performance compared to using constant frequency AIR algorithm. Therefore, error-resilience tools are essential for video communications over noisy wireless channels. If the video source is unable to provide such services, then a suitable placed active node can perform the necessary computations on behalf of the sender. Furthermore, adaptation in the network has the advantage of showing fast response to changing channel conditions and accordingly adjusting the bit-rate and error-resilience.

The transcoding class services can also be utilised in the fixed networks. To illustrate this, the active congestion control scheme was proposed as an alternative traffic conditioning mechanism in for the DiffServ network. The evaluation was performed on a single queue of the edge node. This means that the obtained results can also be generalised to the best-effort forwarding model. The transcoder was placed on an active edge node for regulating the input video traffic streams' bit-rate such that they abide by their TCSs as well as making better utilisation of network bandwidth. When the channel utilisation approached to 100%, packet losses experienced due to the bursty characteristic of the traffic aggregate. Transient buffer overflows resulted in packet losses which were reflected as quality degradations in the received video quality. However, when ACC was applied, the packet losses were almost eliminated and a better utilisation of the available network bandwidth was enabled. In effect, ACC can effectively prevent packet-losses and improve the perceived video quality both during the congestion and non-congestion periods. In summary, ACC can provide prevention of packet loss for both video and other applications, improvement of video quality through better utilisation of the network bandwidth, and salvaging of the video quality during the congestion times.

The active network services should not involve very complex operations so that they do not reduce the efficiency of the other active services executing on the same node. In addition, they are required to executable in real-time. Therefore, a number of measures were taken during the design of the transcoder such that its operation is simplified. In this chapter, the complexity analysis of the error-resilient video transcoder was also introduced in order to prove its suitability for implementation in the active network nodes. The experiments have demonstrated that the transcoding imposes minor increment on the end-to-end delay budget (e.g., around 5ms with the aforementioned hardware specifications). This shows that the transcoder can be operated in processing capable network nodes without hampering the real-time operation of the video-based applications.
The research work presented in this thesis has highlighted the QoS related problems of the present-day networks. Based on these observations, a new networking approach was envisaged that can alleviate such problems. This approach has been motivated by the research activities in the active networking field. The processing capabilities of the envisaged future networks can enable dynamic adaptation of multimedia applications to the requirements of the networks, applications, and users. Such processing is realised through the use of active services which will initially coexist with the popular IP QoS architectures. A number of active services have been developed as the main part of the research work presented in this thesis. Various computer simulations were performed to demonstrate the advantages of enabling a processing capable network. The results have provided the confidence to continue advocating the idea that the active networking approach will shape the future of multimedia communications.

6.3 Future Research Directions

The active services QoS support concept presented in this thesis can be further investigated to provide solutions for a variety of networking problems. In this section a brief description of two possible research areas are introduced.

Adaptive Video Multicast:
Transmission of multimedia information to a wide range and number of users using the multicast technology is a challenging task. Existing video multicast approaches are usually single-rate and do not scale well to the requirements of users with heterogeneous capabilities. Therefore, stream or layer switching based multicast approaches have been proposed to account for the deficiencies of single-rate multicast. These approaches require end-to-end signalling between the client and server in order to change the subscribed multicast group. In stream switching based multicast, the number of available streams are usually very limited (i.e., mostly 3) and is not efficient in terms of bandwidth utilisation. On the other hand, layered multicast can make better use of network resources but is also limited in the number of layers that can be provided. In addition, if the network is unable to sustain enough bandwidth for the base layer, the application may loose its utility. In addition, drift effect may be encountered during the layer switching process if the first frame of the switched layer is not an I-frame. FGS coding solves this problem by always using the base layer as the motion prediction reference to all enhancement layers. However, FGS suffers from coding efficiency.

Three factors are of great importance when designing a video multicast service. These are, the video coding techniques to be used, adaptation mechanisms to be employed within the multicast routers, and the mechanism to detect the link status between the network nodes. In order to satisfy
the user requirements and efficiently utilise the network resources, intelligent network configuration tools, i.e. active network and/or agent technologies, should be used. By deploying transcoding services when and where necessary in the networks, single-rate multicast streams can be adapted to the requirements of individual users without affecting the performance received by the other users. Preliminary investigations performed during the research work presented in this thesis have shown that the proposed transcoding architecture is able to produce multiple streams at different rates with an increased transcoding delay of around 2.5ms for every other output video stream. Research in this area is related to potential benefits of employing active networking technologies both from the network operator and the end-user point of view.

**Pervasive Computing Support for Multimedia QoS:**

In future, users will be surrounded by pervasive computing devices capable of providing advanced multimedia services. It is envisaged that the QoS provision in such networking architecture will be based on active network notion at the delivery network side. This will be complemented by ad-hoc networking of pervasive devices around the user terminal for advanced QoS support in short-range communications. Deploying active routers within the delivery network can alleviate the bandwidth and channel restrictions by executing active services, i.e. content adaptation mechanisms. The ad-hoc control of pervasive devices will allow the use of the surrogate hardware resources and enable users to benefit from multiple applications independent of their terminal capabilities (e.g. the PDA device controlling the surround-sound HI-FI for playing music, using a PC-monitor for video display and its own screen for web-browsing). This research will focus on providing end-to-end support for multimedia applications by introducing innovative self-configuring and self-organising network services. This will be realised in two forms: Firstly, *active services for content adaptation* for basic service provision. Secondly, *application-aware offloading* for surrogate resource usage and *service mobility* for seamless migration of applications between the surrounding pervasive devices for advanced service provision.

**Active services for content adaptation:** The delivery network contains high-power active/programmable nodes for multimedia processing. In order to comply with the service level agreements between the user and the content/service provider, applications will be tailored by executing active services as and when necessary. Such services are envisaged to range from application layer multicast, rate-control, spatial resolution adaptation and error-resilience to media format conversions, etc.

**Application-aware offloading:** This is a case of ad-hoc networking where the user acts as the controlling node and distributes the rich multimedia traffic or computation tasks to the
neighbouring devices. Application-aware offloading will allow for an end-client with restricted resources to dynamically offload and distribute parts of its workload to any surrogate device(s) with advanced capabilities in the vicinity. The distribution will comprise partitioning of the media applications and offloading the computationally intensive tasks using the pre-defined complexity models for remote parallel processing. Such operation will help preserve the client resources and enable servicing of rich applications in return with advanced QoS. This can be realised in two different offloading methods:

- **Surrogate caching**: The multimedia information is cached at the surrogate device, personalised and uploaded back to the user terminal on-demand.
- **Process division**: A computationally complex application, in consideration to its complexity model, is segmented into different processes and distributed among other devices for parallel processing.

**Service mobility**: If users decide to follow an application using another pervasive audio/visual device with enhanced facilities, they can route the multimedia stream directly without any intermediate processing at the user terminal. Such migration of applications between surrounding devices is called service mobility. This will provide the flexibility and convenience of selecting the most desired device(s) as well as allowing for multiple applications to be run at various devices simultaneously.
References


[BakeF2]  F. Baker, K. Chan and, A. Smith, “Management information base for the
differentiated services architecture”, RFC 3289, May 2002.

selective packet discard of streamed MPEG video under low bandwidth conditions,”
Proceedings of IEEE International Conference on Multimedia and Expo (ICME


queueing”, Proceedings of IEEE Fifteenth Annual Joint Conference of the IEEE Computer
CA, USA.


Wroclawski, and E. Felstaine, “A Framework for integrated services operation over

1998.

Ph.D. Thesis, Georgia Institute of Technology, Jul. 1999, USA.

network caches”, Proceedings of IEEE Seventeenth Annual Joint Conference of the
IEEE Computer and Communications Societies (INFOCOM 1998), vol. 2, pp. 600-
608, 29 Mar./2 Apr. 1998, San Francisco, CA, USA.


management”, ACM SIGCOMM Computer Communication Review, vol. 30, Issue: 3,

Proceedings on Communications Architectures, Protocols and Applications

Proceedings of the IEEE International Conference on Image Processing (ICIP

measurement and implications for end-to-end QoS”, Proceedings of the 1998 ICPP
Workshops on Architectural and OS Support for Multimedia Applications/Flexible
Communication Systems/Wireless Networks and Mobile Computing, pp. 3-12, 14
Aug. 1998, Minneapolis, MN, USA.


Bibliography


Bibliography

time applications over a differentiated services network”, Proceedings of the Seventh

release, The JNT Association, Jul. 2001,
http://www.ja.net/development/qos/qos_ttt_report.pdf

[UMA] Special Issue on Universal Multimedia Access, IEEE Signal Processing Magazine,

networks: joint effect of encoding rate and data loss on user-oriented QoS”, In the

[VetrA] A. Vetro, C. Christopoulos, and H. Sun, “Video transcoding architectures and

streaming”, Proceedings of the IEEE International Conference on Image

algorithms for real-time video distribution”, IEEE/ACM Transactions on Networking,

[VitoF] F. De Vito, L. Farinetti, and J. C. De Martin, “Perceptual classification of mpeg
video for differentiated services communications”, Proceedings of IEEE
International Conference on Multimedia and Expo (ICME 2002), vol. 1, pp. 141-144,

[VogeA] A. Vogel, B. Kerherve, G.V. Bochman, and J. Gecsei, “Distributed multimedia and


compression”, IEEE Transactions on Circuits and Systems for Video Technology,

Watanabe, “Video transcoding proxy for 3Gwireless mobile internet access”, IEEE

[WethD] D. Wetherall, “Active network vision and reality: lessons learned from a capsule-
based system”, Proceedings of DARPA Active Networks Conference and Exposition

network bandwidth reservation scheme”, Proceedings of IEEE Second Conference
on Open Architectures and Network Programming (OPENARCH 1999), pp. 59-66,
26-27 Mar. 1999, New York, NY, USA.


215


