Project no: 507613
Project acronym: Euro-NGI
Project title: Design and Engineering of the Next Generation Internet, towards convergent multi-service networks.

Instrument: Network of Excellence (NoE)
Thematic Priority: Information Society Technologies (IST)

**Deliverable reference number:** D.WP.JRA.2.2.1

**Deliverable title:**
TRAFFIC MANAGEMENT IN A MULTI-PROVIDER CONTEXT: PRELIMINARY REPORT, STATE OF THE ART

Due date of deliverable: 2004/05/31
Actual submission date: 2004/05/31

Start date of project: 1st December 2003  Duration: 3 years

Organisation name of lead contractor for this deliverable: Inesc-ID
Editor’s name for this deliverable: Paulo Rogério Pereira

Revision 1

---

**Project co-funded by European Commission within the Sixth Framework Programme (2002-2006)**

<table>
<thead>
<tr>
<th>Dissemination Level</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PU</td>
<td>Public</td>
</tr>
<tr>
<td>PP</td>
<td>Restricted to other programme participants (including the Commission Services)</td>
</tr>
<tr>
<td>RE</td>
<td>Restricted to a group specified by the consortium (including the Commission Services)</td>
</tr>
<tr>
<td>CO</td>
<td>Confidential, only for members of the consortium (including the Commission Services)</td>
</tr>
</tbody>
</table>
With contributions of the following partners:

<table>
<thead>
<tr>
<th>Partner Number</th>
<th>Partner Name</th>
<th>Contributor Name</th>
<th>Contributor e-mail address</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>GET</td>
<td>David Ros</td>
<td><a href="mailto:David.Ros@irisa.fr">David.Ros@irisa.fr</a></td>
</tr>
<tr>
<td>10</td>
<td>INRIA</td>
<td>Bruno Tuffin</td>
<td><a href="mailto:Bruno.Tuffin@irisa.fr">Bruno.Tuffin@irisa.fr</a></td>
</tr>
<tr>
<td>11</td>
<td>FT</td>
<td>Jim Roberts</td>
<td><a href="mailto:james.roberts@rd.francetelecom.com">james.roberts@rd.francetelecom.com</a></td>
</tr>
<tr>
<td>12</td>
<td>PRISM</td>
<td>Loubna Echabbi</td>
<td><a href="mailto:Loubna.Echabbi@prism.uvsq.fr">Loubna.Echabbi@prism.uvsq.fr</a></td>
</tr>
<tr>
<td>12</td>
<td>PRISM</td>
<td>Madiagne Diallo</td>
<td><a href="mailto:Madiagne.Diallo@prism.uvsq.fr">Madiagne.Diallo@prism.uvsq.fr</a></td>
</tr>
<tr>
<td>27</td>
<td>TNG-Polito</td>
<td>Claudio E. Casetti</td>
<td><a href="mailto:casetti@polito.it">casetti@polito.it</a></td>
</tr>
<tr>
<td>27</td>
<td>TNG-Polito</td>
<td>Marco Mellia</td>
<td><a href="mailto:mellia@polito.it">mellia@polito.it</a></td>
</tr>
<tr>
<td>29</td>
<td>TLC.Group-Pisa</td>
<td>Gregorio Procissi</td>
<td><a href="mailto:g.procissi@iet.unipi.it">g.procissi@iet.unipi.it</a></td>
</tr>
<tr>
<td>29</td>
<td>TLC.Group-Pisa</td>
<td>Michele Pagano</td>
<td><a href="mailto:m.pagano@iet.unipi.it">m.pagano@iet.unipi.it</a></td>
</tr>
<tr>
<td>29</td>
<td>TLC.Group-Pisa</td>
<td>Rosario Garroppro</td>
<td><a href="mailto:r.garroppo@iet.unipi.it">r.garroppo@iet.unipi.it</a></td>
</tr>
<tr>
<td>29</td>
<td>TLC.Group-Pisa</td>
<td>Stefano Giordano</td>
<td><a href="mailto:s.giordano@iet.unipi.it">s.giordano@iet.unipi.it</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Anne Nevin</td>
<td><a href="mailto:annene@Q2S.ntnu.no">annene@Q2S.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Bjørnar Libæk</td>
<td><a href="mailto:libak@q2s.ntnu.no">libak@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Odd Inge Hillestad</td>
<td><a href="mailto:hillesta@q2s.ntnu.no">hillesta@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Peder J. Emstad</td>
<td><a href="mailto:peder@q2s.ntnu.no">peder@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Peter Svensson</td>
<td><a href="mailto:svensson@q2s.ntnu.no">svensson@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Stian Johansen</td>
<td><a href="mailto:stianjo@q2s.ntnu.no">stianjo@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Victor Nicola</td>
<td><a href="mailto:nicola@q2s.ntnu.no">nicola@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>32</td>
<td>NTNU</td>
<td>Yuming Jiang</td>
<td><a href="mailto:jiang@q2s.ntnu.no">jiang@q2s.ntnu.no</a></td>
</tr>
<tr>
<td>36</td>
<td>WUT</td>
<td>Andrzej Beben</td>
<td><a href="mailto:abeiben@tele.pw.edu.pl">abeiben@tele.pw.edu.pl</a></td>
</tr>
<tr>
<td>36</td>
<td>WUT</td>
<td>Marek Dabrowski</td>
<td><a href="mailto:mdabrow@tele.pw.edu.pl">mdabrow@tele.pw.edu.pl</a></td>
</tr>
<tr>
<td>36</td>
<td>WUT</td>
<td>Halina Tarasiuk</td>
<td><a href="mailto:halina@tele.pw.edu.pl">halina@tele.pw.edu.pl</a></td>
</tr>
<tr>
<td>36</td>
<td>WUT</td>
<td>Wojciech Burakowski</td>
<td><a href="mailto:wojtek@tele.pw.edu.pl">wojtek@tele.pw.edu.pl</a></td>
</tr>
<tr>
<td>37</td>
<td>Inesc-ID</td>
<td>Paulo Rogério Pereira</td>
<td><a href="mailto:prbp@inesc.pt">prbp@inesc.pt</a></td>
</tr>
<tr>
<td>39</td>
<td>CEMAT</td>
<td>Nelson Antunes</td>
<td><a href="mailto:nantunes@ualg.pt">nantunes@ualg.pt</a></td>
</tr>
<tr>
<td>40</td>
<td>UPB</td>
<td>Bogdan Malureanu</td>
<td><a href="mailto:bogdan.malureanu@orange.ro">bogdan.malureanu@orange.ro</a></td>
</tr>
<tr>
<td>40</td>
<td>UPB</td>
<td>Eugen Borcoci</td>
<td><a href="mailto:eugen.borcoci@elcom.pub.ro">eugen.borcoci@elcom.pub.ro</a></td>
</tr>
</tbody>
</table>
Summary

1. INTRODUCTION .................................................................................................................. 6

2. TAXONOMY .......................................................................................................................... 7

   2.1. QoS REQUIREMENTS AND CLASS OF SERVICE DIFFERENTIATION .................. 7
       2.1.1. QoS Definition ........................................................................................................ 7
       2.1.2. QoS Requirements .................................................................................................. 8
       2.1.3. Service Differentiation ............................................................................................ 12

   2.2. QUEUING MECHANISMS ............................................................................................ 15
       2.2.1. Requirements .......................................................................................................... 16
       2.2.2. What aspects impact on the requirements queuing mechanisms should fulfil? .... 17
       2.2.3. FIFO queuing .......................................................................................................... 18
       2.2.4. Fair queuing ............................................................................................................ 18
          2.2.4.1. Generalized Processor Sharing ................................................................. 18
          2.2.4.2. Weighted Round Robin .................................................................................. 18
          2.2.4.3. Deficit Round Robin ....................................................................................... 19
          2.2.4.4. Weighted Fair Queuing .................................................................................. 19
       2.2.5. Delay-controlled queuing ....................................................................................... 20
          2.2.5.1. Waiting Time Priority .................................................................................... 20
          2.2.5.2. Advanced Waiting Time Priority ............................................................... 20
          2.2.5.3. Earliest Due Date ............................................................................................ 20
          2.2.5.4. Weighted Earliest Due Date ......................................................................... 21
       2.2.6. Rate-controlled queuing ......................................................................................... 21
       2.2.7. Class-based queuing ............................................................................................... 21
          2.2.7.1. Priority Queuing ............................................................................................... 21
          2.2.7.2. Custom queuing ............................................................................................... 22
          2.2.7.3. Modified Deficit Round Robin ....................................................................... 22
          2.2.7.4. Class Based Weighted Fair Queuing ........................................................... 23
          2.2.7.5. Duplicate Scheduling with deadlines ......................................................... 23
          2.2.7.6. Priority Scheme based on Queue Management Algorithm with Reservations 24

   2.3. QoS AND PRICING ....................................................................................................... 25
       2.3.1. QoS and pricing: a discussion of two contrasting approaches ................. 25
       2.3.2. Relationship between QoS and pricing mechanisms ....................................... 26
       2.3.3. A network provider point of view on pricing and QoS ........................................ 28
          2.3.3.1. Return on investment ..................................................................................... 28
          2.3.3.2. Price discrimination ....................................................................................... 28
          2.3.3.3. Congestion pricing ......................................................................................... 29
          2.3.3.4. Conclusions .................................................................................................... 31

3. SELECTED TOPICS ............................................................................................................... 35

   3.1. INTER DOMAIN ADMISSION CONTROL ................................................................. 35
       3.1.1. Inter-domain architecture and assumed IDAC scenario .................................... 36
       3.1.2. Appropriate characterization of traffic profile ...................................................... 38
          3.1.2.1. Evaluation of traffic profile deformation and recalculation of TD ............ 39
          3.1.2.2. Numerical results .......................................................................................... 42
             3.1.2.2.1. Evaluation of traffic profile deformation ................................................. 43
             3.1.2.2.2. Evaluation of IDAC approach .............................................................. 45
       3.1.3. Summary and open issues ..................................................................................... 47

   3.2. ADMISSION CONTROL FOR SLA MANAGEMENT ............................................ 48

   3.3. SLA/SLS MANAGEMENT IN MULTI-DOMAIN ENVIRONMENT ................................ 51

       3.3.1. Introduction ........................................................................................................... 51
       3.3.2. Business Models .................................................................................................. 52
       3.3.3. Policy Based Network Management Concepts ................................................ 52
       3.3.4. Service Definitions, Service Level Agreements ................................................ 54
       3.3.5. QoS classes of services ....................................................................................... 55
       3.3.6. QoS guarantees .................................................................................................... 56
3.3.7. Intra-domain aspects
  3.3.7.1. Architectures
  3.3.7.2. Service (SLA/SLS) subscription and invocation
  3.3.7.3. Resource and Traffic Management
  3.3.7.4. Specific protocols

3.3.8. Inter-domain aspects
  3.3.8.1. QoS peering methods
  3.3.8.2. Architectures and protocols
  3.3.8.3. Service subscription and invocation
  3.3.8.4. Resource and Traffic Management

3.3.9. Examples of relevant architectures

3.4. "CLASSICAL" VS. CHAOTIC-BASED PREDICTIONS
  3.4.1. Classical Predictors
    3.4.1.1. LMMSE Predictor
    3.4.1.2. LMS Predictor
    3.4.1.3. Normalized LMS (NLMS) Predictor
    3.4.1.4. LMK Predictor
    3.4.1.5. Geometric-Predictors

  3.4.2. Chaotic Predictors
    3.4.2.1. Background on chaotic systems
      3.4.2.1.1. Strange Attractors
    3.4.2.2. The prediction problem
    3.4.2.3. Metrics for quality of prediction
    3.4.2.4. Prediction algorithms
      3.4.2.4.1. Approximation techniques
      3.4.2.4.2. Linear Local Predictor
      3.4.2.4.3. Radial Basis Functions Predictor

3.5. SERVICE DIFFERENTIATION FOR AUDIO AND VIDEO STREAMS
  3.5.1. QoS Requirements
    3.5.1.1. QoS Requirements for Audio
    3.5.1.2. QoS Requirements for Video

  3.5.2. Audio and Video End System
    3.5.2.1. Audio
    3.5.2.2. Video and Compression Techniques
    3.5.2.3. Error control for video coding
    3.5.2.4. DiffServ Network and Video Coding

  3.5.3. Traffic Characterization
    3.5.3.1. Traffic Models
    3.5.3.2. Open Challenges

  3.5.4. Admission Control
    3.5.4.1. Open Challenges

3.6. PRICING MECHANISMS
  3.6.1. Flat-rate pricing
  3.6.2. Paris Metro pricing (PMP)
  3.6.3. Edge pricing
  3.6.4. Effective bandwidth pricing
  3.6.5. Expected capacity pricing
  3.6.6. Priority pricing
  3.6.7. Smart-market pricing and auctions
  3.6.8. Responsive pricing
  3.6.9. Proportional Fairness Pricing
  3.6.10. A restricted bi-level pricing
  3.6.11. Pricing for guaranteed performance services

4. OPEN PROBLEMS
  4.1. BANDWIDTH SHARING BETWEEN ELASTIC FLOWS WITH UNRESPONSIVE TRAFFIC
  4.2. TRAFFIC CHARACTERIZATION
  4.3. ADMISSION CONTROL
  4.4. INTER DOMAIN ADMISSION CONTROL

5. CONCLUSIONS
List of Figures

Figure 2.1: Effects of absolute end-to-end delay by the E-model. ................................................................. 9
Figure 2.2: Packet classification and traffic conditioning. ............................................................................. 13
Figure 2.3: An implementation example of the WRR mechanism. ................................................................. 19
Figure 2.4: Modified Deficit Round Robin. ..................................................................................................... 23
Figure 2.5: QoS mechanisms, pricing mechanisms and time scales [Reichl 03]. ............................................... 27
Figure 3.1: Inter-domain admission control scenario. ..................................................................................... 36
Figure 3.2: General model of multi-domain network. ...................................................................................... 37
Figure 3.3: Delay and loss function block. ....................................................................................................... 38
Figure 3.4: Survey of possible solutions to the problem of traffic profile deformation. ................................... 39
Figure 3.5: Illustration of traffic profile deformation. ..................................................................................... 40
Figure 3.6: Monitoring delay characteristics by active probing. .................................................................... 41
Figure 3.7: The studied system. ..................................................................................................................... 42
Figure 3.8: Difference between $b'$ and $b$ vs. the number of routers in a chain, $m$. ......................................... 44
Figure 3.9: Ratio of $b'/b$ vs. the ratio of flow peak rate to the link capacity. ..................................................... 45
Figure 3.10: Ratio of $e'/e$ vs. the number of routers in a chain, $m$. ................................................................. 46
Figure 3.11: Prediction scheme. ...................................................................................................................... 69
Figure 3.12: LMS algorithm scheme. ............................................................................................................... 71
List of Tables

Table 2.1: QoS levels related to ODP viewpoints ................................................................. 8
Table 2.2: Effects of one-way end-to-end delay. ................................................................. 9
Table 2.3: Static properties of a communications task. ....................................................... 10
Table 2.4: Application QoS metrics. .................................................................................. 11
Table 2.5: QoS targets for audio and video applications. .................................................. 11
Table 2.6: QoS targets for data applications. ................................................................. 12
Table 2.7: Techniques for achieving good QoS............................................................... 13
Table 2.8: QoS Classes...................................................................................................... 15
Table 3.1: Packet loss ratio experienced by flows submitted to link L, assuming unmodified and recalculated TDs of flow #2 ................................................................. 47
Table 3.2: Audio Formats. ............................................................................................... 81
Table 3.3: Video Formats. ............................................................................................... 82
1. Introduction

Deliverable DJRA.2.2.1 is the first deliverable of work-package JRA.2.2 on traffic management in a multi-provider context, providing a state of the art on the related subjects. This document was jointly edited by the participating partners, promoting partner integration and being the base for further research within this workpackage.

Traffic management concerns the way network resources are shared between competing users. Traffic flows can be broadly classified as streaming flows and non-streaming or data flows. Streaming flows require a certain bandwidth, maximum delay, jitter or packet loss. Non-streaming flows, on the other hand, are elastic, being tolerant to varying network conditions.

This document analyses the state of the art on concepts and techniques necessary to a fair sharing of network resources according to the quality of service requirements of the traffic flows generated by each user application.

The structure of the deliverable is as follows. Chapter 2 provides a taxonomy of important concepts of traffic management, such as quality of service, quality of service requirements of different types of applications, service differentiation, queuing mechanisms and pricing. Chapter 3 provides a deeper review of some hot research topics, such as inter domain admission control, service level agreements and their relationship with admission control, "classical" vs chaotic-based predictions for obtaining information on the future behaviour of network traffic, service differentiation of audio and video streams, and a survey of pricing schemes. Chapter 4 lists some open problems that will be focused by research in the near future. Chapter 5 presents the conclusions. At the end, a list of abbreviations is provided to help understanding the document.
2. Taxonomy

This chapter provides a taxonomy of important concepts of traffic management. The first section starts by defining quality of service, then analyses the quality of service requirements of different types of applications, describes how traffic can be classified in classes and some basic techniques for providing quality of service. The next section discusses queuing mechanisms that can be used to control the share of network resources offered to each traffic class for controlling its quality of service. Finally, the last section discusses the relation between quality of service and pricing.

2.1. QoS Requirements and Class of Service Differentiation

2.1.1. QoS Definition

Quality of Service (QoS) is defined by the International Telecommunication Union [E.800] as "the collective effect of service performance which determine the degree of satisfaction of a user of the service". A similar concept, defined in the same recommendation, is Network Performance, which is "the ability of a network or network portion to provide the functions related to communications between users". Thus, Network Performance is the technical part of QoS, while QoS is the user perspective of the network performance.

The Internet community also has different abstraction level views of QoS [RFC 3198]. At a high level of abstraction, QoS refers to the ability to deliver network services according to the parameters specified in a Service Level Agreement - an agreement between the user and the service provider. In this definition, "Quality" is characterised by parameters such as service availability, delay, jitter, throughput and packet loss ratio. On the other hand, at a network resource level, QoS refers to a set of capabilities that allow a service provider to prioritise traffic, control bandwidth, and network latency. These are the functions that allow the network to offer the users the quality they expect. A more complete description and comparison of different IP QoS definitions is given in [Gozdecki 03].

QoS requirements differ according to the abstraction level, being more qualitative and general at the user level, and more quantitative and technical at the systems level. It is possible to have several QoS abstraction levels. Table 2.1 [Leydekkers 96] shows a possible classification of QoS levels according to the Open Distributed Processing (ODP) Reference Model's viewpoints [X.903]. For each ODP viewpoint, this classification shows: the QoS level, how the QoS is perceived by users; and examples of parameters describing how the
QoS is implemented in the equipment. Each ODP viewpoint models a particular aspect of system characteristics. The enterprise viewpoint focuses on the purpose, scope and policies of a system. The information viewpoint defines the semantics of information and the semantics of information processing in a system. The computational viewpoint deals with the functional decomposition of a system into objects which interact at interfaces, allowing a distributed structure of the system. The engineering viewpoint focuses on mechanisms and functions required to support distributed interaction between objects in a system. Finally, the technology viewpoint is concerned with the choice of technology for implementing the system.

<table>
<thead>
<tr>
<th>ODP Viewpoint</th>
<th>QoS level</th>
<th>Characteristics</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enterprise</td>
<td>Subjective QoS</td>
<td>End-user oriented QoS requirements (not formal and subjective)</td>
<td>“Audio must be along the video”; “Video should be similar to television-quality”</td>
</tr>
<tr>
<td>Information</td>
<td>Objective QoS</td>
<td>Precise statement of QoS requirements derived from the subjective QoS specification. Often application independent</td>
<td>25 fps (PAL), 30 fps (NTSC) Aspect ratio of 4:3 (PAL TV Format) Telephone Quality: 8 KHz</td>
</tr>
<tr>
<td>Computational</td>
<td>Application QoS</td>
<td>Describe the QoS requirements for the applications often specified in terms of media quality and media relations</td>
<td>Lip synchronisation between audio and video can be expressed in ms. (e.g. synchronisation skew ± 80 ms)</td>
</tr>
<tr>
<td>Engineering</td>
<td>System QoS</td>
<td>Describe the QoS requirements from the operating system</td>
<td>Buffer size, Operations/s, Memory (Mb)</td>
</tr>
<tr>
<td>Network</td>
<td>QoS</td>
<td>Describe the QoS requirements from the network</td>
<td>Throughput (Mbit/s), Jitter (ms), Delay (ms)</td>
</tr>
<tr>
<td>Technology</td>
<td>QoS properties</td>
<td>Describe the QoS characteristics of devices, operating system and network technologies. QoS properties.</td>
<td>Video device (PAL format), Audio device (µ-law) Cell loss rate for ATM</td>
</tr>
</tbody>
</table>

Table 2.1: QoS levels related to ODP viewpoints.

The next subsection (2.1.2) will examine the QoS requirements the users expect and those that the network may offer, while the following subsection (2.1.3) will examine mechanisms that allow the network to offer different levels of QoS to different user applications.

In this document, the Internet community's definition of QoS is used, adding adjectives, if necessary, to distinguish the different QoS views.

### 2.1.2. QoS Requirements

At a user level, QoS may be subjective or objective [Cochrane 91]. Subjective QoS is estimated by the service provider in terms of the user opinion, collected by user surveys. Objective QoS is measured in terms of parameters which are expressed in a user
understandable language appropriate to the concerned service, and which are user verifiable. An example is provided in [G.114], that uses the E-model, which is based on subjective tests of delay (among other parameters), to estimate the effects of delay on mouth-to-ear speech transmission quality, as shown in figure 2.1. These results lead to the definition of three classes of delay, as shown in table 2.2. Accordingly, it was recommended not to exceed a one-way delay of 400 ms [Y.1541], and to keep the delay below 150 ms whenever possible [G.114].

![Figure 2.1: Effects of absolute end-to-end delay by the E-model.](image)

<table>
<thead>
<tr>
<th>One-way delay</th>
<th>Effects of one-way end-to-end delay</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 150 ms</td>
<td>most applications, both speech and non-speech, will experience essentially transparent interactivity</td>
</tr>
<tr>
<td>150 - 400 ms</td>
<td>some users dissatisfied</td>
</tr>
<tr>
<td>&gt; 400 ms</td>
<td>nearly all users dissatisfied</td>
</tr>
</tbody>
</table>

Table 2.2: Effects of one-way end-to-end delay.

The applications can be decomposed in a set of communication tasks. Table 2.3 [F.700] presents a non exhaustive list of communication tasks' static properties. These properties may help to characterise the communication and help identify QoS requirements.
<table>
<thead>
<tr>
<th>Property</th>
<th>Possible values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Communication configuration</td>
<td>Point-point/point-multipoint/multipoint-point/multipoint-multipoint</td>
</tr>
<tr>
<td>Symmetry of information flow</td>
<td>Unidirectional/bidirectional-symmetric/bidirectional-asymmetric</td>
</tr>
<tr>
<td>Transmission control entity</td>
<td>Source/sink/source and sink/third party</td>
</tr>
<tr>
<td>Communication delay</td>
<td>Real-time&lt;br&gt;Near-real-time&lt;br&gt;Non-real-time&lt;br&gt;Specified time</td>
</tr>
<tr>
<td>Mandatory media components</td>
<td>Audio/video/text/still picture/graphics/data</td>
</tr>
<tr>
<td>Optional media components</td>
<td>Audio/video/text/still picture/graphics/data/none</td>
</tr>
<tr>
<td>Media component interrelations</td>
<td>1) Synchronization between:&lt;br&gt;a) audio and video (lip synchronism, location related synchronism);&lt;br&gt;b) audio and text (voice synthesis);&lt;br&gt;c) text and video/still picture/graphics (subtitles synchronized with images);&lt;br&gt;d) graphics and audio.&lt;br&gt;2) Symmetry between media components of the same type to allow for bidirectionality. 3) Conversion between information types (or media components).</td>
</tr>
<tr>
<td>Time continuity</td>
<td>Isochronous/non-isochronous</td>
</tr>
</tbody>
</table>

Table 2.3: Static properties of a communications task.

The application's perception of QoS parameters may be affected by a number of application-level performance metrics [Miras 02] shown in table 2.4. Recommendation [X.641] provides a more complete list of QoS characteristics of general importance to communications and processing, in the following groups: time-related, coherence, capacity-related, integrity-related, safety-related, security-related, reliability-related and other QoS characteristics.

According to the type of application, different QoS requirements may be required. Table 2.5 [G.1010] presents QoS requirements for audio and video applications, while table 2.6 [G.1010] presents QoS requirements for data applications. From these requirements, applications can be classified as loss tolerant or loss intolerant, and in one of four general areas of delay tolerance: interactive (delay << 1s), responsive (delay ~ 2s), timely (delay ~ 10s) and non-critical (delay >> 10s).
<table>
<thead>
<tr>
<th><strong>QoS metric</strong></th>
<th><strong>Description</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Throughput</td>
<td>Effective share of bandwidth that the application is getting from the network.</td>
</tr>
<tr>
<td>Delay or Latency</td>
<td>End-to-end delay that the application experiences.</td>
</tr>
<tr>
<td>Delay variation (Jitter)</td>
<td>If the end-to-end delay does not remain constant, temporal inconsistency may happen.</td>
</tr>
<tr>
<td>Data loss</td>
<td>Amount of information the user perceives as missing. Can depend on packet loss, packet loss pattern and error correction and concealment techniques.</td>
</tr>
<tr>
<td>Service Availability</td>
<td>Describes the requirement for uninterrupted service with acceptable quality.</td>
</tr>
<tr>
<td>Security</td>
<td>Privacy and trusted communication service.</td>
</tr>
</tbody>
</table>

**Table 2.4: Application QoS metrics.**

<table>
<thead>
<tr>
<th><strong>Medium</strong></th>
<th><strong>Application</strong></th>
<th><strong>Degree of symmetry</strong></th>
<th><strong>Typical data rates</strong></th>
<th><strong>Key performance parameters and target values</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>One-way delay</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>&lt;150 ms preferred (Note 1)</td>
</tr>
<tr>
<td>Audio</td>
<td>Conversational voice</td>
<td>Two-way</td>
<td>4-64 kbit/s</td>
<td>&lt; 1 ms</td>
</tr>
<tr>
<td>Audio</td>
<td>Voice messaging</td>
<td>Primarily one-way</td>
<td>4-32 kbit/s</td>
<td>&lt; 1 s for playback</td>
</tr>
<tr>
<td>Audio</td>
<td>High quality streaming audio</td>
<td>Primarily one-way</td>
<td>16-128 kbit/s (Note 3)</td>
<td>&lt;&lt; 1 ms</td>
</tr>
<tr>
<td>Video</td>
<td>Videophone</td>
<td>Two-way</td>
<td>16-384 kbit/s</td>
<td>&lt; 10 s</td>
</tr>
<tr>
<td>Video</td>
<td>One-way</td>
<td>One-way</td>
<td>16-384 kbit/s</td>
<td>&lt; 10 s</td>
</tr>
</tbody>
</table>

**Table 2.5: QoS targets for audio and video applications.**

**NOTE 1** – Assumes adequate echo control.
**NOTE 2** – Exact values depend on specific codec, but assumes use of a packet loss concealment algorithm to minimise effect of packet loss.
**NOTE 3** – Quality is very dependent on codec type and bit-rate.
**NOTE 4** – These values are to be considered as long-term target values which may not be met by current technology.
<table>
<thead>
<tr>
<th>Medium</th>
<th>Application</th>
<th>Degree of symmetry</th>
<th>Typical amount of data</th>
<th>Key performance parameters and target values</th>
<th>One-way delay (Note)</th>
<th>Delay variation</th>
<th>Information loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data</td>
<td>Web-browsing – HTML</td>
<td>Primarily one-way</td>
<td>~10 KB</td>
<td>Preferred &lt; 2 s /page</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acceptable &lt; 4 s /page</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Bulk data transfer/retrieval</td>
<td>Primarily one-way</td>
<td>10 KB-10 MB</td>
<td>Preferred &lt; 15 s</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acceptable &lt; 60 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Transaction services – high priority e.g. e-commerce</td>
<td>Two-way</td>
<td>&lt; 10 KB</td>
<td>Preferred &lt; 2 s</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acceptable &lt; 4 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Command/ control</td>
<td>Two-way</td>
<td>~1 KB</td>
<td>&lt; 250 ms</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Still image</td>
<td>One-way</td>
<td>&lt;100 KB</td>
<td>Preferred &lt; 15 s</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acceptable &lt; 60 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Interactive games</td>
<td>Two-way</td>
<td>&lt;1 KB</td>
<td>&lt; 200 ms</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Telnet</td>
<td>Two-way (asymmetric)</td>
<td>&lt; 1 KB</td>
<td>&lt; 200 ms</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>E-mail (server access)</td>
<td>Primarily one-way</td>
<td>&lt;10 KB</td>
<td>Preferred &lt; 2 s</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Acceptable &lt; 4 s</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>E-mail (server to server transfer)</td>
<td>Primarily one-way</td>
<td>&lt;10 KB</td>
<td>Can be several minutes</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Fax (&quot;real-time&quot;)</td>
<td>Primarily one-way</td>
<td>~10 KB</td>
<td>&lt; 30 s/page</td>
<td>N.A.</td>
<td>10^-6 BER</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Fax (store &amp; forward)</td>
<td>Primarily one-way</td>
<td>~10 KB</td>
<td>Can be several minutes</td>
<td>N.A.</td>
<td>10^-6 BER</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Low priority transactions</td>
<td>Primarily one-way</td>
<td>&lt;10 KB</td>
<td>&lt; 30 s</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>Usenet</td>
<td>Primarily one-way</td>
<td>Can be 1 MB or more</td>
<td>Can be several minutes</td>
<td>N.A.</td>
<td>Zero</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NOTE – In some cases, it may be more appropriate to consider these values as response times.

Table 2.6: QoS targets for data applications.

2.1.3. **Service Differentiation**
There are several techniques to achieve good QoS [Tanenbaum 03], as listed and described in table 2.7. Some of these techniques, such as admission control, resource reservation and packet scheduling will be detailed in later sections of this document, while others techniques like routing will be addressed in other work packages of this project.
<table>
<thead>
<tr>
<th>Technique</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overprovisioning</td>
<td>Increase router capacity, buffer space and bandwidth so that packets fly</td>
</tr>
<tr>
<td></td>
<td>through easily.</td>
</tr>
<tr>
<td>Buffering</td>
<td>Flows can be buffered on the receiving side before being delivered. Buffering</td>
</tr>
<tr>
<td></td>
<td>does not affect the reliability or bandwidth, and increases the delay, but</td>
</tr>
<tr>
<td></td>
<td>smooths out the jitter.</td>
</tr>
<tr>
<td>Traffic shaping</td>
<td>Regulates the average rate at the sending side to smooth the traffic.</td>
</tr>
<tr>
<td>Resource reservation</td>
<td>Make sure the bandwidth, buffer space and CPU cycles are available for the</td>
</tr>
<tr>
<td></td>
<td>traffic on its route to the destination.</td>
</tr>
<tr>
<td>Admission Control</td>
<td>Make decisions about which traffic flows can be accepted.</td>
</tr>
<tr>
<td>Proportional Routing</td>
<td>Instead of using only the best path to the destination, divide the traffic</td>
</tr>
<tr>
<td></td>
<td>among the available routes.</td>
</tr>
<tr>
<td>Packet Scheduling</td>
<td>Packets should be sent not in the order of arrival to a queue, but according</td>
</tr>
<tr>
<td></td>
<td>to the QoS required.</td>
</tr>
</tbody>
</table>

Table 2.7: Techniques for achieving good QoS.

Due to the price that some of these solutions require, it is important to classify the traffic flows according to the QoS required, and apply some of these techniques only according to the QoS required by each flow. An example is overprovisioning, which is the simplest and most used technique. By overprovisioning the network, the network capacity is increased so that an adequate QoS level is provided. The problem with this solution is that all traffic flows get the same QoS level, as required by the most demanding applications, even if they do not need such a good QoS level. This usually results in a significant waste of network resources, making the other solutions economical alternatives.

The traffic classification can be made for each flow when it starts, in what is called admission control. Alternatively, the classification can be made per packet.

Packets are classified, marked and shaped according to a scheme similar to what is shown in figure 2.2 [RFC 2475].

![Packet classification and traffic conditioning diagram](image-url)
Packet classifiers select packets in a traffic stream, "steering" the packets matching some specified rule to an element of traffic conditioning for further processing. The classification is usually based on information from the packet header, such as source address, destination address, differentiated services field, protocol ID, source port and destination port numbers, and other information such as incoming interface or hour of the day. Policies [RFC 3060][RFC 3460] can be used as rules for the classification process.

The classification can be made by one of the following methods [Kilkki 99]:
1. The users selects a definite service class from the available classes.
2. The application automatically selects a preferable service class for each flow or packet.
3. The network selects an appropriate service class based on information about the application.
4. The network selects an appropriate service class based on the customer contract regardless of the application.
5. A combination of the first four approaches.

A traffic conditioner may contain the following elements: meter, marker, shaper and dropper. A traffic profile specifies the temporal properties of a traffic stream selected by a classifier. The traffic profile provides rules for determining whether a particular packet is in-profile or out-of-profile. For example, a profile could be based on a token bucket with rate \( r \) and burst size \( b \). In this example, out-of-profile packets are those packets in the traffic stream which arrive when insufficient tokens are available in the bucket. A meter is used (where appropriate) to measure the traffic stream against a traffic profile. The state of the meter with respect to a particular packet (e.g., whether it is in- or out-of-profile) may be used to affect a marking, dropping, or shaping action. Markers assign an importance level to each packet, marking it on the packet header. Shapers delay some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. Droppers discard some or all of the packets in a traffic stream in order to bring the stream into compliance with a traffic profile. This process is know as "policing" the stream.

The traffic flows will get different QoS according to the classification made. There are several class categories, as described in table 2.8, in descending order of quality. Guaranteed services have some non-technical deployment problems [Teitelbaum 02], like: poor incremental deployment properties, intimidating new complexity for network operators, missing functionality on routers, and serious economic challenges. This raised the interest in lower than best effort services, that are easier to integrate in the current network infra-structure.
<table>
<thead>
<tr>
<th>Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Guaranteed</td>
<td>These classes provide deterministic QoS guarantees. Examples are: the</td>
</tr>
<tr>
<td></td>
<td>guaranteed service [RFC 2212] of Integrated Services, the Expedited</td>
</tr>
<tr>
<td></td>
<td>Forwarding [RFC 3246] of Differentiated Services, and the CBR and VBR</td>
</tr>
<tr>
<td></td>
<td>services of ATM.</td>
</tr>
<tr>
<td>Better than Best Effort</td>
<td>These classes provide soft or statistic QoS guarantees. Examples are:</td>
</tr>
<tr>
<td></td>
<td>Assured Forwarding [RFC 2597].</td>
</tr>
<tr>
<td>Alternative Best Effort</td>
<td>Although no QoS guarantee is provided, classes are provided with low delay</td>
</tr>
<tr>
<td></td>
<td>and high drop rate, or low drop rate and high delay as in [Hurley 01].</td>
</tr>
<tr>
<td>Best Effort</td>
<td>This is the default Internet class, that provides no QoS guarantee.</td>
</tr>
<tr>
<td>Lower than Best Effort</td>
<td>These classes get a level of service that is potentially degraded compared</td>
</tr>
<tr>
<td></td>
<td>to the default best effort service. An example is the Internet2 QBone</td>
</tr>
<tr>
<td></td>
<td>Scavenger Service (QBSS) [Shalunov 01].</td>
</tr>
</tbody>
</table>

**Table 2.8:** QoS Classes.

The QoS classes of table 2.8 are what is called a relative model, since usually the only guarantee is that the traffic of a high-priority class will receive no worse service than that of a low priority class. An alternative model is a quantitative schema: the proportional QoS model [Chen 03][Christin 03]. The proportional model provides adjustable and consistent differentiation between classes. An example of proportional packet delay specification is: "Class-2 Delay $\approx 4 \times$ Class-1 Delay". It is possible to have combinations of proportional packet delay, proportional packet jitter, proportional packet loss and absolute loss rates, delays, or throughput.

### 2.2. Queuing Mechanisms

In packet switched networks, queuing mechanisms (also named schedulers, service disciplines, queuing disciplines, …) are located in the routers and correspond to the packet handling level, so operate at the time scale from μs to ms. They allow us for re-ordering of arriving packets to a router before their transmission by the outgoing links. As a consequence, the order of packets sending by the router can be changed (even radically) comparing with the packet arrivals. In this way, the queuing mechanisms can support, when congestion arises, a prioritising service of selected packet streams, e.g. belonging to the same flow, by queuing or dropping packets from other streams. By doing re-ordering of packets, different objectives referring to quality of service at the packet level can be achieved, corresponding to e.g. packet delays, packet losses, link sharing between concurrent flows. However, to fulfil a specified objective requires adequate type of queuing mechanism. This is a reason why so many types of queuing mechanisms have been considered and new proposals are still submitted. Therefore, choosing appropriate queuing mechanism is an important issue.
In this part, we survey several queuing mechanisms that are proposed in the literature, which could be adopted for supporting service performances in packet networks. We start with defining the main requirements, that should be considered in designing process of queuing mechanisms. Furthermore, we point out on main aspects having impact on defining a group of requirements corresponding to particular queuing mechanism. Finally, we provide a brief description of particular groups of queuing mechanisms: fair queuing, delay-controlled queuing, rate-controlled queuing, and class-based queuing.

2.2.1. Requirements

Designing process of queuing mechanisms meets four main requirements and, finally, results to a trade-off between them [Keshav 97]. These requirements are:

- Ease of implementation: this requirement is mostly important from the scalability issue, especially in high-speed networks.
- Fairness and protection: in the context of queuing mechanisms by fairness we mean fair link bandwidth and buffer sharing between concurrent flows. We call a resource allocation fair if it satisfies the max-min fair share criterion [Keshav 97]. While protection means that misbehaviour by one flow should not affect performance received by other flows. The relationship between fairness and protection is that a fair share queuing mechanism automatically provides protection, because it limits a misbehaving flow to its fair share. However, the converse need not be true.
- Performance bounds: this requirement is significant for the networks which offer services with predefined QoS guarantees. In this case, from the packet level mechanisms we expect packet handling according to the accepted QoS requirements, which may be expressed by the following QoS parameters: admissible packet delay, packet jitter, packet losses, or link bandwidth. Queuing mechanisms play a significant role for supporting these requirements in the network. In addition, queuing mechanisms should also support service differentiation (if needed) between packet streams belonging to different classes of services.
- Ease and efficiency of admission control (AC): the importance of this requirement is for the networks offering QoS guarantees and requiring implementation of AC function. From the AC viewpoint, estimation of available resources (as link bandwidth, buffer) for particular service classes is strongly required to be done in a simple way. However, the queuing mechanism should not lead to network under-utilisation.
2.2.2. **What aspects impact on the requirements queuing mechanisms should fulfil?**

Network architecture, types of traffic and QoS provided by the network constitute the main factors having great impact on the requirements for a queuing mechanism.

We distinguish between two main types of traffic in IP-based networks, that are streaming and elastic. They differ in both traffic characteristics and QoS requirements. Streaming traffic is rather packet delay sensitive and is produced by real-time applications (as voice, video), while elastic traffic is rather packet loss sensitive and is emitted by non-real time applications (as HTTP, FTP). To meet the QoS requirements for the above types of traffic, the queuing mechanisms should provide different service for different packet streams.

Since network architecture may be flow-aware or class-aware or mix of both, therefore depending on the assumed concept we can apply flow-aware or class-aware queuing mechanisms or both types. A type of queuing mechanism is determined by network node tasks mainly. While flow-aware queuing mechanisms dedicate a separate queue to packets from single flow (connection), class-aware dedicate a separate queue to packets coming from a group of flows (aggregated flow), each associated with the same class of service. In the class-aware scheme the queuing mechanism provides different quality of service to different classes, while flows within a class affect the same service quality. Of course there are advantages and disadvantages of applying flow-aware or class-aware queuing mechanisms in the network nodes. The main advantage of the class-aware queuing is in reducing the state in the queuing mechanism (such as pointers to packet queues, and memory about the service already received by a single flow), which is the critical resource in implementing this mechanism. Thus, the smaller the amount of queuing mechanism state required for a particular queuing discipline, the easier it is to implement. On the other hand the main problem with aggregation is that flows within the same class are not protected from each other. Because the scheduler cannot distinguish between connections in the same class, the misbehaviour of one connection in the class affects the whole. A second problem with aggregation is that if a queuing mechanism gives aggregated congestion feedback signal to an ensemble of feedback flow-controlled connections, a well behaved connection may perceive congestion signals caused by the bad behaviour of other connections as its own misbehaviour. However, if the traffic sharing a class is well behaved and flows sharing a class have approximately the same traffic characteristics, then aggregating them into a class does not significantly affect the degree of protection [Keshav 97].

Taking into account a kind of QoS the network can offer for packet handling, we point out strict (deterministic or statistical), relative (e.g. proportional to the assigned priority/weights)
QoS guarantees, and best effort networks without any QoS guarantees. A network architecture supporting strict QoS guarantees needs adequate packet level mechanisms. Furthermore it should be enhanced by adequate admission control (AC) mechanisms defined at a flow level. The network architecture with such mechanisms can handle traffic by guarantying strict defined QoS parameters. Contrary to the architectures which fulfil strict QoS requirements, the relative QoS usually do not require any AC. On the other hand best effort networks tend to serve flows in a fair manner.

2.2.3. **FIFO queuing**
FIFO (First In First Out) is the simplest queuing mechanism. In this mechanism, the incoming packets are transmitted to the output queue in the order they arrive, and are dropped when packets arrive to a full queue. According to the FIFO discipline all packets are handle by the same queue with the same priority and as a consequence the mechanism cannot differentiate among flows or class of service. This mechanism is also known as FCFS (First Come First Served). FCFS is applied in current best effort networks.

2.2.4. **Fair queuing**
We can distinguish between five queuing mechanisms, which try to fulfil fairness requirements. They are, Generalised Processor Sharing (GPS), Weighted Round Robin (WRR), Deficit Round Robin (DRR), and Weighted Fair Queuing (WFQ). We call these mechanisms fair queuing mechanisms.

2.2.4.1. **Generalized Processor Sharing**
GPS is ideal work-conserving fair queuing mechanism. It serves packets as if they are in separate logical queues, visiting each non empty queue in turn and serving an infinitesimally small amount of data from each queue, so that in any finite time interval it can visit every logical queue at least once. Additionally, queues can be associated with service weights, and they receive service in proportion to these weights, whenever they have data in the queue. In this case GPS fulfil also max-min weighted fair share [Keshav 97]. We can say that GPS fulfil fairness requirements in any finite time interval. Unfortunately, GPS is practically unimplementable. However, GPS is the reference queuing mechanism for evaluating how other queuing mechanisms approximate fairness requirements.

2.2.4.2. **Weighted Round Robin**
The simplest approximation of GPS is Round Robin (RR) mechanism, which serves a packet from each non empty queue, instead of an infinitesimally small amount of data. It well
approximates GPS if all queues have the same service weights and all packets are of the same size. In the case of different weights an extension of RR is WRR. Figure 2.3 shows an implementation example of WRR mechanism with three queues which differ in service weights. In this case we assume that every queue is permanently occupied by packets which are of the same size. If packets from different queues are of different sizes, WRR divides each weight by mean packet size of a flow associated with the respective queue to assign a packet service priority. As a consequence, WRR needs a knowledge of mean packet sizes of handling flows. In practice, it is very difficult for realizing. Comparing with GPS, WRR provides fairness rather in long time scale. In the case of the same packet sizes, this time scale is equal to round length, but could be longer if packets are of different sizes.

![Diagram of WRR mechanism](image)

**Figure 2.3:** An implementation example of the WRR mechanism.

**2.2.4.3. Deficit Round Robin**

DRR modifies WRR to allow for handling packets of variable sizes without the knowledge of the mean packet size corresponding to each flow. On the contrary to the simple WRR algorithm, DRR takes deficit of bytes into account and it is not needed to specify byte counts based on mean packet sizes. Thanks to that, implementation of DRR is easier than WRR.

**2.2.4.4. Weighted Fair Queuing**

Methods which well approximate the GPS mechanism are Weighted Fair Queuing (WFQ) and Packet-by-packet Generalized Processor Sharing (PGPS). However, both mechanisms do not divide a packet into an infinitesimally small amount of data as GPS and they do not require the knowledge of the mean packet size as WRR. Since the results of both mechanisms are the same we will focus on WFQ only, which is not so difficult for implementing like PGPS. On the other hand, there are many methods for realizing WFQ. In general, WFQ simulates the GPS scheme and simulation results fix the order of packet service. In practice, WFQ calculates packet time stamps instead of packet time service. The main difference between WFQ methods consists in rules to obtain packet time stamps. Comparing with WRR or DRR, WFQ is the mechanism which approximates GPS in the best way. It supports fairness
requirements even in short time scale. On the other hand, the implementation of WFQ is rather complex since it requires calculating time stamps for all queuing packets and sorting output queues according to the obtained time stamps. There are some variants for implementing WFQ. One of them is Self-Clocked Fair Queuing (SCFQ) [Golestani 94], a second one is Start-Time Fair Queuing (STFQ) [Goyal 96].

2.2.5. Delay-controlled queuing
Several queuing mechanisms have been proposed that provide a relative differentiation with respect to mean packet delays. The well-known mechanism is Waiting Time Priority (WTP), which is also enhanced to Advanced Waiting Time Priority (AWTP). On the other hand, there were also proposals of some queuing mechanisms for providing strict delay differentiation, like Earliest Due Date (EDD) and Weighted Earliest Due Date (WEDD).

2.2.5.1. Waiting Time Priority
WTP is a priority mechanism in which the priority of a packet increases in proportion to its waiting time, and it is known as the best queuing mechanism to achieve the proportional delay differentiation model in short time scales [Dovrolis 99a, Dovrolis 99b, Dovrolis 02]. A separate queue is dedicated to packets associated with each priority class. Moreover, every queue is associated with a priority factor. When a packet is admitted in the router, the date is stored in it. This date permits to compute the waiting time of a packet when a packet is at the head of the queue. WTP mechanism uses waiting times and priority factors to determine a priority (normalised waiting times) of the head-of-queue packets. The packet with the highest priority is served first. The WTP mechanism was first studied by L. Kleinrock under the name Time-Dependent Priorities [Kleinrock 76].

2.2.5.2. Advanced Waiting Time Priority
In [Lai 03], authors propose advanced WTP mechanism - AWTP. Comparing with WTP, AWTP additionally takes into account packet transmission time to determine a packet priority. Simulation results provided in [Lai 03] show that AWTP not only obtains more accurate delay proportion than WTP, but also reduces the average waiting time.
A disadvantages of both WTP and AWTP mechanisms is that they only provide delay differentiation based on average delays.

2.2.5.3. Earliest Due Date
EDD mechanism provides so called strict delay differentiation. In this mechanism each service class $i$ is associated with a delay bound $d_i$. A packet of class $i$ arriving at time $t_A$
receives a tag $t_i + d_i$ representing its deadline. The packets to be forwarded are scheduled in increasing order of their deadlines. The packets with expired deadlines are dropped. There are two extensions of EDD mechanism, Delay-EDD and Jitter-EDD [Zhang 95]. Delay-EDD specifies the process by which the mechanism assigns deadlines to packets. During call set-up each source negotiates a service contract with the mechanism. In Jitter-EDD mechanism, a delay-jitter regulator precedes the EDD scheduler. With a delay-jitter regulator, all packets receive the same delay at every hop (except at the last hop), so the difference between the largest and the smallest delays, which is the delay-jitter along the connection, is reduced to the delay-jitter on the last hop.

2.2.5.4. Weighted Earliest Due Date

WEDD enhances the EDD mechanism in such a way that not only different delay bounds but also different deadline violation probabilities are provided [Bodamer 00]. The ratio of violation probabilities in different classes are specified by a set of parameters.

2.2.6. Rate-controlled queuing

Rate controlled queuing mechanisms are a group of mechanism that give connection bandwidth, delay and jitter bounds. A rate controlled queuing mechanism consist of a regulator and a scheduler. The regulator delays packets until they are eligible, and the scheduler arbitrates among eligible packets. By choosing different regulators and schedulers, a rate-controlled mechanism can provide a variety of bandwidth, delay, and delay-jitter guarantees [Keshav 97].

2.2.7. Class-based queuing

The class-based queuing concept is investigated in [Floyd 95]. The main reason to consider such an approach is to provide priority link sharing between aggregated flows corresponding to classes of service defined by a network provider. One option for such a mechanism is to limit link bandwidth of the regulated class, while a second option is to decrease the priority of the regulated class.

2.2.7.1. Priority Queuing

In Priority Queuing (PQ), each queue is associated with a priority level. If there are $n$ priority levels and a higher–number priority level corresponds to a queue with higher priority, the mechanism serves a packet from priority $k$ only if there are no packets awaiting service in levels $k+1, k+2, ..., n$. Selecting a packet for transmission depends only on the number of priority levels. The problem with the PQ mechanism is that some low priority flows may be
blocked or starved. Packets in higher priority class get served first. Packets in a lower class get served only if all other higher queues are empty. It means that higher priority queues yield lowest delay, highest throughput and bandwidth. Even though, it does not allow end-to-end guarantees to be provided on a per-flow basis. It only provides for one class of traffic to receive better service than the other traffic classes which share the same link. Thus, in PQ mechanism, it is critical that appropriate AC and policing restrict the service rates at higher priority levels.

2.2.7.2. Custom queuing
Custom queuing is simply based on the WRR scheme. Contrary to WRR mechanism presented in the context of the fair queuing mechanism, in this case queues are associated with service classes instead of single flows. As a consequence, the number of queues is limited to the number of classes. The custom queuing concept is provided by CISCO.

2.2.7.3. Modified Deficit Round Robin
MDRR provides special support for delay sensitive traffic on CISCO 12000 GSR series routers. MDRR includes a high priority queue that is treated differently from the other queues associated with service classes. Within MDRR, the queues are served in round-robin fashion except for one, the special queue served with high priority. MDRR extends DRR adding high priority queue. For DRR each queue has assigned configurable value called a service quantum. A service quantum provides a measure of how much traffic should be handled from the queue in each round. Packets from that queue are served until their cumulative length (byte count) exceeds the service quantum. A deficit counter is used as a credit mechanism. The deficit counter value is added to the service quantum to determine the measure of service available for each queue during each round. The high priority queue can be served in two modes: strict priority or alternate priority. In first mode, high priority queue is served first until all of its packets are sent and the queue is empty, then the remaining queues are served according to the DRR scheme. In the alternate priority mode, packets are served first from high priority queue then from the active DRR queue. The active queue is selected in round-robin fashion. Figure 2.4 depicts general model of MDRR.
2.2.7.4. **Class Based Weighted Fair Queuing**

The idea of Class Based Weighted Fair Queuing is very similar to MDRR. In CBWFQ a functionality of DRR mechanism is replaced with the WFQ mechanism. Since implementation of WFQ is rather complex issue CBWFQ is implemented in edge routers rather than in high speed network nodes. CBWFQ mechanism is also called Priority Queuing Waited Fair Queuing (PQWFQ) [Bak03].

2.2.7.5. **Duplicate Scheduling with deadlines**

Duplicate Scheduling with Deadlines (DSD) is investigated in [Hurley 01]. DSD supports so called Alternative Best Effort (ABE) service. In ABE service packets are marked either green or blue. DSD is based on the concept of duplicates. Duplicates of all incoming packets are sent to a virtual queue with buffer size $\text{Buff}$. A duplicate is admitted if virtual buffer is not full. Packets in the virtual queue are served according to FCFS at rate $C$, as they would be in flat best effort. The times at which duplicates will be served are used to assign blue packets deadlines at which they would have (approximately) been served in flat best effort. The original arriving packets are fed according to their colour into a green and blue queue. Deadlines are assigned to packets as they arrive, green and blue packets are queued separately, and the deadlines of the packets at the head of blue and green queues are used to determine which is to be served next. Blue packets are always served at the latest their deadline permits subject to work conservation. Green packets are served in the mean time if they have been in the queue for less than $d$ s, and dropped otherwise. Some of the most important properties of DSD are the following. (1) Buffer space constraint: the total buffer occupancy for real packets is always less than $\text{Buff}$, the size of virtual queue used for duplicates. (2) All accepted blue packets will be served by their deadlines. Accepted blues are
just served at the same time as, or earlier than, they would have been in flat best effort. (3) All green packets are served before \( d \), or otherwise dropped. Low bounded (per hop) delay for the green packets is enforced by dropping a green packet that waits or would have to wait \( d \) seconds in the queue.

DSD mechanism is recommended to be implemented at network periphery, where bit rates are on order of a few megabits per second since they introduce non-negligible queuing delays.

### 2.2.7.6. Priority Scheme based on Queue Management Algorithm with Reservations

Priority Scheme based on Queue Management Algorithm with Reservations (PS-QMAR) is described in [Burakow 00, Burakow 01]. This queuing mechanism is designed to support so-called Green Line service, which provides a relative service differentiation between \( N \) priority levels. PS-QMAR is fed by \( N \) independent packet flows each served with different priority level. Priority level should be associated with an aggregate packet flow. Furthermore, PS-QMAR assumes that for the flow-1 has been assigned the highest priority (priority-1) while for the flow-\( N \) the lowest (priority-\( N \)). PS-QMAR is based on the QMAR algorithm. In this algorithm, respective queue place reservations are made for the packets belonging to the flows from 1 to (\( N-1 \)), only. For each flow, a single reserved place (for incoming packet) in the queue is maintained. Then, the total number of place reservations in the considered system is constant and equal to (\( N-1 \)). According to the QMAR algorithm, an arriving packet (belonging to the flows from 1 to (\( N-1 \)) seizes the reserved place dedicated for it (not necessary the last one in the current queue). Furthermore, the reservation for next incoming packet of this flow is made, in a way depending on the priority level. The reserved place is moving up to the top of the queue according to the system service process and is holding even in the case it reaches the top of the queue (before arrival of next packet). When it takes place, the arriving packet finds its reservation on the head of the queue and, then is served as the first. The desired priority differentiation for \( N \) flows in the PS-QMAR is get by application of appropriate queue place reservation rules for the particular flows. Better service quality of the \( i \)-th flow over the \( j \)-th flow (\( i,j = 1,\ldots, N-1 \); \( i<j \)) is obtained by using the rule that the reservation for the \( i \)-th flow is always placed before the reservation for the \( j \)-th flow. Since the \( N \)-th flow is served with the lowest priority, no reservation is considered for it. As a consequence better service quality means that in each router with PS-QMAR mechanism mean delay of packets belonging to the flows served with highest priorities should be smaller.
and not greater than mean delay of packets belonging to the flows served with lower priorities.

2.3. **QoS and Pricing**

Quality of service (QoS) issues in packet networks have been the subject of a great deal of research effort. However, it is safe to say that many of the solutions studied in the literature (like those described in the previous sections) deal only with the *technical* aspects of offering different levels of QoS guarantees to packet flows. Indeed, *economic*-related issues, such as charging and accounting, were neglected by most of the networking community until a few years ago. In this section we will focus in the relationship between pricing and QoS; a survey of pricing schemes will be the subject of Section 3.6. Some of the arguments presented here are somewhat controversial and reflect the fact that no consensus exists yet among networking researchers.

2.3.1. **QoS and pricing: a discussion of two contrasting approaches**

Pricing in IP networks is usually regarded as a tool serving several purposes [Courcoubetis 03]:
1. Allocation of network resources, according to the users’ willingness to pay.
2. Recovering operational and expansion costs of the network.
3. Congestion Control.

Items 2 and 3 above will be discussed in detail in Section 2.3.3 from the point of view of an Internet Service Provider.

Regarding Item 1, there has been much debate in the networking community concerning the “right” approach to deal with QoS issues in IP networks\(^1\), and the related problem of *charging for QoS*. Essentially, the traditional approach has been centred on adding QoS architectures, protocols and mechanisms to the current best-effort Internet. It is a commonly-held view that offering distinct classes of service makes it necessary to apply different charges to customers (flows) of each class [Cocchi 93]. The classical, simple “flat-rate” pricing scheme [Anania 97] is deemed as inappropriate for a multi-service network with diverse QoS classes. This is so because, in the absence of price discrimination, all users would always request the “best” treatment, potentially leading to a degradation of the quality of the highest service class. In this context, *usage-based pricing* arises naturally. Pricing schemes adapted to QoS

---

\(^1\) The issue of QoS in IP networks is a complex one, and the debate continues to this day.
architectures in IP networks have been the subject of recent research (see e.g. [DaSilva 00] and [Blefari 03], and references therein).

Nevertheless, the jury is still out on whether such tools will be necessary for building a multi-service Internet. Proponents of the “brute force” approach to QoS, which consists in overprovisioning the network, claim that congestion is no longer (or will no longer be) an issue—hence, that there are neither technical nor economical justifications for altering the current best-effort model of the Internet. Remark that an implicit assumption of QoS architectures like DiffServ [RFC 2475] is that network resources (typically bandwidth) are scarce: such mechanisms can therefore be regarded as a means to avoid congestion by appropriately managing scarcity. However, network bandwidth is becoming a commodity and transmission capacity (at least, in the backbone of the Internet) has increased dramatically, with plenty of unused optical fibre as a consequence of massive investments carried out in the late 90’s [Odlyzko 01]. Since the cost of transporting the bits is very low, the problem of offering an improved QoS—i.e., of reducing the probability of congestion—is solved by simply increasing the bandwidth. Some authors, like Odlyzko [Odlyzko 00], state that flat-rate pricing is then a natural consequence of overprovisioning in IP networks. Odlyzko suggests segmenting the market for network access in a few tiers at most, with users in each service class being treated in a best-effort manner.

Critics of the “brute force” approach to QoS contend that, in spite of the commoditization of network capacity, there still are some problems with that approach, since congestion has simply “moved” from the core of the network to its edges. For instance, wireless networks often contain one or more bottlenecks, as is the case in the upcoming UMTS, with link speeds limited to a few hundreds of kbit/s per terminal. Hence, QoS architectures—and their corresponding pricing schemes—may still be needed. Besides, even if (according to anecdotal evidence [Odlyzko 01]) users prefer simple pricing schemes like flat-rate, the experimental results obtained by the INDEX project [Edell 99][Altmann 01b] suggest that users are also willing to pay a premium, in the form of an usage-based additional charge, to have an on-demand access to a better service class.

2.3.2. Relationship between QoS and pricing mechanisms

Since users are supposed to be cost-sensitive, pricing is often seen as a traffic engineering tool, allowing network operators to have an influence on user behaviour and, therefore, on the

---

1 The following quote from a recent workshop on QoS in IP networks [RIPQOS 03] summarises well the debate: “There’s a sense that something might be needed [in order to have QoS], but little agreement on why and who will pay. At times the very notion of QoS has seemed to be a pointless waste of time, almost a solution waiting for a problem.”
quality of service offered by the network. For instance, *time-of-day pricing* is frequently used to try to evenly distribute user demand over time. On a shorter time scale, some pricing schemes aim at giving feedback to users in times of congestion by means of increased prices (*congestion pricing*), so that users react by reducing their demand, alleviating the congestion. The following figure, adapted from [Reichl 03], illustrates the relationship between QoS mechanisms and pricing mechanisms, in terms of the different *time scales* in which they operate.

![Figure 2.5: QoS mechanisms, pricing mechanisms and time scales [Reichl 03].](image)

We can see that pricing schemes acting on short time scales can be associated to QoS mechanisms like those found in the IntServ [RFC 1633] and DiffServ [RFC 2475] architectures. Such short-term mechanisms usually have interesting economic properties, such as *incentive compatibility* (i.e., selfish users will naturally tend to act in a way that is considered as “beneficial” for the network); however, the complexity of some of these mechanisms make them difficult to implement.

We may establish the following classification of pricing mechanisms.

**Dynamic versus static pricing.** Pricing schemes like flat-rate pricing can be said to be *static* in the sense that prices are constant (at least over long time scales) and, most of all, independent of the instantaneous state of the network. Remark that time-of-day pricing can be seen as a static mechanism since prices are easily predictable by the user. Note also that a (short-time scale) usage-based tariff with a fixed per-volume charge can also be considered as static. On the other hand, in *dynamic* schemes congestion events result in prices that vary over time, reflecting the fluctuations in network load. From the point of view of users, lack of predictability may be an important drawback of dynamic schemes.

**Pricing with or without resource reservation.** In IP networks, and depending on the QoS architecture and the service level chosen, per-flow service guarantees can be strict or statistical in nature. Accordingly, pricing schemes should be adapted to the kind of guarantees offered by the network. For instance, in reservation-based services offering strong QoS
guarantees, like IntServ’s *Guaranteed Service*, the quality of service requirements—and, hence, the tariff—are negotiated at the admission control phase.

2.3.3. **A network provider point of view on pricing and QoS**

In this section we discuss the relationship between pricing and quality of service *from the point of view of a commercial network provider*. This relationship is often obscured by the dual role of pricing: to assure return on investment and to control congestion.

### 2.3.3.1. Return on investment

Return on investment is the prime objective of the network provider. It is necessary that levied charges cover all the capital and operational costs of running the network. Prices of different items (e.g., connection charge, modem rental, usage charges) should be somewhat related to the costs incurred but there remains considerable flexibility in the way they are attributed.

The cost of an individual IP flow is difficult to evaluate. It is not appropriate to use the marginal cost of handling its packets since this is arguably negligible. It is more a question of devising a means for sharing overall network costs in an appropriate way. Considerable work on exactly how this should be done has been performed in the context of telephone network interconnection charges (e.g., see [Baumol 94]). Long run average incremental costs are frequently used to determine interconnection charges.

A similar formalism is not necessary for the unregulated Internet but the way interconnection charges are evaluated does show that even traffic handled by otherwise idle resources still incurs a cost and is susceptible to charging. A reasonable assumption is that the cost of a flow is proportional to the volume of data transmitted. The cost may also depend on the burstiness of the flow or on whether it is streaming or elastic. However, these considerations are of secondary importance as they arguably have a negligible impact on provisioning.

### 2.3.3.2. Price discrimination

Cost is not the only factor determining price. In particular, price discrimination is economically efficient when there exist distinct market segments with different willingness to pay for basically the same service. Many different devices can be employed as a key to discrimination. The airline industry is a useful reference. Business class comfort justifies a price difference with tourist class that largely exceeds the difference in cost. Further price
discrimination is practiced in tourist class by the weekend stay-over clause which allows leisure travellers to pay less than business travellers for exactly the same quality of service.

The need for price discrimination in the Internet is often identified with a need to offer distinct QoS classes. Unfortunately, our understanding of the way QoS depends on traffic volume and characteristics suggests it is not easy to create a networking equivalent to business class and tourist class. QoS guarantees through traffic contracts (for individual flows or traffic aggregates) are only advantageous in situations of overload. This, of course, may be a useful distinction if overloads are frequent or have very serious consequences when they do occur. However, there is no means to ensure that a premium service is manifestly and consistently better than best effort in the same way that business class is better than tourist class.

Fixing the price of a traffic contract is problematic. The cost of a flow depends ultimately on the volume of data emitted, and not on the traffic parameters declared in the traffic contract. One must, therefore, question the long term sustainability of charging based on a contractual traffic descriptor.

This distinction between streaming and elastic traffic might constitute a key to price discrimination. However, willingness to pay is not systematically greater for real time audio and video flows than for data transfers. The per-byte transport cost of both types of traffic is roughly the same.

Alternative keys to price discrimination are probably more acceptable than necessarily vague QoS guarantees. For example, the speed of a DSL modem is a significant price factor in current networks. There is also considerable scope for service bundling and the design of specific pricing packages. These alternatives can effectively segment the market in the same way that the weekend stop-over clause segments the market for tourist class air travel.

2.3.3.3. Congestion pricing

Most research on network pricing is concerned with congestion control and not return on investment. The best known example of congestion pricing is the “smart market” proposed by MacKie-Mason and Varian [MacKie 95]. In the smart market, users include a bid in each packet. In case of congestion, the users offering the lowest bids are discarded first and accepted packets are priced at a rate determined by the highest bid among the rejected packets.

From this example, it is clear that congestion pricing is not concerned with return on investment. When the network is not congested, there is no charge so that a well provisioned
network gains no revenue from the smart market. The objective is rather to optimally share a scarce resource by inciting users to reveal their utility and attributing the resource those who gain the most.

The smart market was proposed more as an illustration of the principle of congestion pricing than as a practical system. A more pragmatic approach was advanced by Shenker et al. [Shenker 96]. These authors suggest that it is sufficient to offer differentially priced service classes with charges increasing with the guaranteed level of quality of service. Users regulate their charge by choosing or not to use a higher quality of service class in times of congestion. A proposal along the same lines using Diffserv classes of service was recently advanced by Shu and Varaiya [Shu 03].

Kelly has proposed an alternative congestion pricing framework [Kelly 00]. His “self managed networking” scheme is based on a reactive congestion control like that of TCP where explicit congestion notification (ECN) marks are issued to signal imminent congestion. Each mark received by the user implies a unit charge. In the event of congestion, users with high utility continue their transmissions. They receive more marks and pay a surcharge but successfully complete their transaction. Users with low utility will refrain from transmitting until the congestion ceases.

Despite the popularity of the above schemes in the networking research community, there are serious reservations on the use of congestion pricing by a commercial network operator. In the first place, network resources are generally not scarce. The provider can easily upgrade capacity and will do so before congestion occurs if return on investment is assured. Congestion may then be interpreted by users as a sign of bad management. Since other charges must already cover network cost, users might find it unreasonable to pay extra when bad planning or bad maintenance results in congestion.

It is difficult to find examples of other service industries where congestion pricing is successfully employed. Most, like the telephone network, use pricing to share overall costs. They ensure by provisioning that congestion occurs rarely.

Congestion in the telephone network is manifested by blocking. The use of admission control ensures admitted traffic is completed in good conditions. This appears as a natural condition for the application of simple usage-based charging: the network sells a service that is always of adequate quality; if demand temporarily exceeds supply, not all customers can be satisfied; however, only satisfied customers have to pay.
Experience in the commercial Internet and similar service industries shows that customers have a very strong preference for simplicity and risk avoidance [Odlyzko 01][Altmann 01]. It is unlikely on these grounds alone that they would ever accept the unpredictability of congestion pricing. Complexity is also an issue for the provider whose operating costs are significantly lower with a simple volume-based charging scheme.

2.3.3.4. Conclusions

Our conclusions on pricing from the provider point of view are as follows:

– return on investment must be assured by appropriately sharing network capital and operating costs between users;
– price discrimination is economically efficient but should be based on criteria other than pretended QoS guarantees;
– congestion pricing, used to efficiently share a scarce resource, is not a satisfactory charging basis for a commercial network operator;
– user preference for simplicity and transparency can be satisfied by a simple volume-based charging scheme in a network equipped with admission control.

These conclusions will have to be taken into account when appraising dynamic pricing schemes whose provable microeconomic optimality may not necessarily be viable for a commercial network provider.

references


3. **Selected Topics**

This chapter provides a deeper review of some hot research topics. The first section presents inter domain admission control. The following section analyses admission control techniques for SLA management. The next section reviews "classical" vs chaotic-based predictions for obtaining information on the future behaviour of network traffic. The following section covers service differentiation for audio and video streams: audio and video QoS requirements, techniques for maximizing perceived quality, models for traffic characterization, and admission control schemes to ensure newly admitted flows can receive the required service guarantees. The next section offers a survey of pricing schemes. Finally, the last section analyses the relationship between service level specifications and admission control.

3.1. **Inter Domain Admission Control**

Currently, the lack of end-to-end QoS (Quality of Service) solution in the Internet is the main reason why new attractive applications cannot be deployed. While it can be regarded that there are some successful prototype implementations of a single domain IP QoS network, like the European project AQUILA [BBF03], the solution for supporting end-to-end QoS in a multi-domain network, like the Internet, is still a challenging task.

We assume that each domain offers several network services, dedicated for handling specific type of traffic with adequate QoS objectives. This can be achieved by implementing in each domain a Resource Controller (RC) responsible for admitting flows and performing resource management within the single domain (see e.g. AQUILA approach [BBF03]). The intra-domain AC function admits or rejects new flow requests based on the knowledge of available resources within the domain and flow requirements [RMV95][BBD02][BDF03]. The recognized approaches assume calculating the effective bandwidth, which is a measure of the amount of resources needed for handling the flow, taking into account the Traffic Descriptor (TD) declared by a user, usually in form of the parameters of single or double token bucket mechanism.

For providing the end-to-end QoS in multi-domain network, an additional mechanism is required, which we call the Inter-Domain Admission Control (IDAC). For effective performing the IDAC one has to deal with the following problems:

1. Architectural issues (including scalable signaling mechanism)
2. Appropriate characterization of traffic profile
3. Inter-working (mapping) between services offered in different domains
4. QoS splitting/assembling

In the following sub-sections, we present some preliminary studies, concerning the above-mentioned issues. Open problems and further work are summarized at the end of the section.

3.1.1. Inter-domain architecture and assumed IDAC scenario

So far, some of the issues related with designing an architecture for inter-domain signalling and QoS service negotiation, were studied e.g. in AQUILA, CADENUS and TEQUILA IST projects [ACT]. In the AQUILA project, special focus was put on designing a scalable inter-domain signalling protocol [NSD03]. However, design of the complete solution for guaranteeing end-to-end QoS is quite a complex task and has not been definitively solved.

Below, we discuss the general approach for performing IDAC for multi-domain IP QoS network (see figure 3.1). It assumes that the TD of the flow is successively submitted to RCs of the domains on the end-to-end path.

![Figure 3.1: Inter-domain admission control scenario.](image)

The scenario for establishing the QoS enabled flow is as follows. A user submits the flow setup request, which contains TD and QoS requirements, to its home IDAC entity. The IDAC (see figure 3.1) constitutes an additional layer, implemented on top of the RCs of the domains. It is responsible for handling the user signaling and communicates with its local RC and IDAC entities in neighboring domains. The tasks performed by the IDAC are the following:

1. Forwarding the flow setup request to the RC. The RC is responsible for AC and resource reservation inside its domain. Notice, that in the case when different domains use different schemes for description of traffic profile and QoS requirements, the IDAC has to provide an adequate mapping in order to assure a consistent service on the end-to-end basis. If the RC successfully admits a flow within its domain, the IDAC receives a positive acknowledgement and continues with performing the next tasks.
(2) Evaluation of the traffic profile deformation and recalculation of TD. Traffic profile may change due to multiplexing with other traffic inside the domain. Note, that due to stochastic nature of this deformation, after passing the domain the traffic conforms to the recalculated TD only with a certain probability level. This probability should be related with the QoS guaranteed by a considered service.

(3) Assessment of the domain’s contribution to the end-to-end QoS. The flow QoS requirements are expressed in terms of end-to-end parameters, thus they are not directly related with the QoS level offered within a single domain. Therefore, on each step on the end-to-end path the IDAC has to assess the QoS degradation level expected in a particular domain and inform the next domain that it has to offer e.g. lower delay and packet loss ratio, taking into account the degradation introduced in all previously visited domains. This task may be realized with the help of so-called QoS assembling functions, which compose the end-to-end QoS values based on the contributions of consecutive domains. This is also quite complex problem due to variety of QoS parameters and their features. For example, the maximum packet delay is additive while the packet loss ratio is rather multiplicative.

(4) Forwarding the flow setup request, with possibly modified TD and information on the flow QoS requirements, to the IDAC entity in the successive domain. Notice that this task may require some information about the inter-domain routing.

The above-described scheme is repeated until the destination domain is reached. A flow setup request can be blocked in any of the domains on the path if there are no available resources to fulfill the end-to-end QoS requirements.

For providing the IDAC with an abstract view of multi-domain heterogeneous network, the general model is proposed, as depicted on figure 3.2. This model assumes that each domain is represented by a single queue followed by a delay and loss function blocks.

![General model of multi-domain network](image)

**Figure 3.2:** General model of multi-domain network.
The motivation for representing the model as a network of queues is that the resources controlled by AC are always modeled as a link with associated buffer. Therefore, from a point of view of a single flow, the multi-domain network with AC performed in each domain (perhaps even using different algorithms) is always seen as a chain of queues.

The delay and loss function blocks (see figure 3.3) describe the influence of the domain on the analyzed traffic stream. They contain the specification of: (1) the QoS assembling functions, and (2) the TD recalculation functions, which take as input the characteristics and current conditions (e.g. introduced packet delay and loss ratio) of the particular path within the domain. Notice, that since the traffic profile as well as the QoS requirements may be expressed using different parameters, the set of functions present in a particular function block is network- and service- dependent.

![Figure 3.3: Delay and loss function block.](image)

This general model can be applied for any IP QoS network. For example, the DiffServ architecture assumes that the bottleneck is located at the domain ingress, while a core is over-dimensioned. Therefore the AC is performed only on the ingress link, which is directly represented by the queue, while the impact of the core is reflected in functions blocks.

The proposed model can be helpful in designing the details of IDAC mechanism and analysing its performances. For each domain the adequate delay and loss functions have to be defined, which can be done analytically or with the support of measurements. In the following, we focus our attention to the issues related with the design of delay function blocks.

3.1.2. **Appropriate characterization of traffic profile**

Below, we discuss one of the problems, which arise when we perform declaration-based AC in inter-domain network. Due to multiplexing with other traffic inside the domain, the characteristics of traffic flow may change. Thus, the original traffic descriptor, declared by the user and enforced by the policing function at the entry to the first domain, becomes inaccurate and should not be used as a base for AC to all consecutive domains on the end-to-end path.
The above mentioned problem can be quite important in large multi-domain networks, where AC has to be performed in each domain separately. To solve this problem, several solutions are possible, as depicted on figure 3.4.

![Diagram of possible solutions to the problem of traffic profile deformation]

**Figure 3.4:** Survey of possible solutions to the problem of traffic profile deformation.

The first class of solutions is based on replacing the scheduling algorithms on all links inside the network with non-work conserving disciplines, which allow for preserving the profile of the flow. The required feature of such class of schedulers, which we call here “TD-conserving”, is the capability to reconstruct the traffic pattern, represented by particular TD, at each node inside the network. Example of “TD-conserving” discipline is the RCSP scheduler, studied in [ZhFe93]. However, the discussed schedulers require per-flow queuing, keeping the TD of each flow in each node and are quite computationally expensive. Thus, applying them in e.g. IP DiffServ networks does not seem to be feasible.

If work-conserving scheduling disciplines are used, the problem of traffic profile deformation has to be solved in another way. One solution is to perform pure Measurement-Based AC, without using traffic descriptors at all. However, providing strict QoS guarantees can be quite difficult if we base AC decisions only on measurement-based predictions of traffic. Below, we propose another solution, which is based on re-calculation of traffic descriptor. The re-calculation algorithm is supported by measurements.

### 3.1.2.1. Evaluation of traffic profile deformation and recalculation of TD

Figure 3.5 provides the intuitive explanation of the effect of traffic profile deformation. It depicts exemplary periodic ON/OFF source when it enters the network (figure 3.5a), and when it leaves the domain (figure 3.5b). The ON/OFF pattern is convenient for such analysis, since it is precisely and tightly characterized by the token bucket TD, with two parameters: token accumulating rate, $r$, and token bucket size, $b$. As a consequence, such traffic pattern will be the mostly affected by the effect of traffic profile deformation. Assume, that at the
network ingress point the flow was policed by the token bucket mechanism which was properly dimensioned, i.e. there were always enough tokens in it to accommodate the arriving packet. The time-plot of the token counter value is depicted in figure 3.5a as the dotted line.

Now, observe how the traffic emitted within one of the ON periods, starting at time $t_1$ and ending at time $t_2$, has changed while passing the network. Suppose, that the transfer delays of the first and last packet of this ON period were $\tau_1$ and $\tau_2$ respectively, with $\tau_2 < \tau_1$. Assuming that no packets were lost, the total amount of bytes transmitted within the ON period has not changed. However, the burst duration, which at the entry to the network was equal to $(t_2 - t_1)$, got reduced to $(t_2 - t_1) - (\tau_1 - \tau_2)$. The dotted line at the figure 3.5b is the time-plot of the token counter value of the hypothetical token bucket configured with the same parameters $(r,b)$ and running at the entry of the next domain. Notice, that at some point it drops to zero and the shaded part of the burst would not be accommodated. Thus, we may conclude that due to the traffic profile deformation the resource requirements of the flow have increased and should now be expressed by updated token bucket parameters $(r,b')$, where $b' > b$. Notice, that the effect of increasing the duration of some of the bursts is also possible. This may happen if $\tau_2 < \tau_1$. However, since the deterministic token bucket TD has to assume the worst-case behaviour of the flow, it should be dimensioned taking into account the maximum burst reduction.

![Figure 3.5: Illustration of traffic profile deformation.](image)

The related issue of traffic deformation within the access network was discussed in the context of CBR (Constant Bit Rate) connections in ATM networks. The recognized solution was to update the limit of the GCRA (Generic Cell Rate Algorithm, which is the ATM equivalent of the token bucket) with the value of CDVT (Cell Delay Variation Tolerance). A simple method for approximation of CDVT was to replace it by the maximum difference of cell transfer times experienced by two cells belonging to the same connection [RMV95]. Some intuition on this approximation can be obtained from figure 3.5: remark, that the
greatest possible deformation occurs when $\tau_1$ is the maximum and $\tau_2$ is the minimum of observed delays. Thus, $CDVT$ is approximated by $W_{\text{max}} - W_{\text{min}}$, where $W_{\text{max}}$ is the $1-\varepsilon$ quantile and $W_{\text{min}}$ is the minimum of the cell transfer times in the access network. As a “rule of thumb”, $1-\varepsilon$ should be equal to the target probability of transmitting non-conforming cells. Below, we discuss the application of the simple approximation method for recalculation of TD inside the IP QoS domain. Remark, that the aim is to define the delay function block for the inter-domain network model introduced in section 3.1.1. Applying the functions specified in this block, the IDAC should be able to recalculate the TD of the flow before passing the request to the neighbor domain.

The quantile-based $IPDV$ ($IP$ Packet Delay Variation) [$Y1540$] should be measured between the ingress Edge Router, and the egress Border Router (see figure 3.6). The network measurement and monitoring system collects a sample of $IPTD$ ($IP$ Packet Transfer Delay) values, experienced by probe packets emitted within a predefined observation window. The following statistical parameters should be obtained from the collected sample: $IPTD_{\text{min}}$, which is simply the minimum value, corresponding to the constant delay on a given path, and $IPTD_{\text{upper}}$, defined as $1-\varepsilon$ quantile (e.g. with $\varepsilon=10^{-3}$) of the distribution of random variable representing the $IPTD$. More specifically,

$$IPTD_{\text{upper}} = \sup \{ w \mid \Pr \{ IPTD \leq w \} \leq 1 - \varepsilon \} \tag{1}$$

Below, we recall the simple and efficient procedure for estimating $IPTD_{\text{upper}}$ from measurements. It assumes ordering the set of the collected measured values, $IPTD_i$, $i=1,...,n$, into a non-decreasing sequence $X_i$, $i=1,...,n$. The values of $X_i$ are called the order statistics, and the estimate of the $1-\varepsilon$ quantile is $\hat{x}_{1-\varepsilon} = X_{[n(1-\varepsilon)]}$, where $[y]$ denotes the integer ceiling of the real number $y$. Notice, that according to the definition of sequence $X_i$, the proportion of...
values within the sample that are not greater than \( \hat{x}_{\epsilon-\epsilon}, \) is exactly equal to \( 1-\epsilon. \) Thus, Then we can replace \( \hat{IPTD}_{upper} \) with the estimate:

\[
\hat{IPTD}_{upper} = X_{[\epsilon(1-\epsilon)]}
\]

(2)

The \( IPDV \) estimate is then:

\[
\hat{IPDV} = \hat{IPTD}_{upper} - IPTD_{min}
\]

(3)

In analogy with dimensioning GCRA in ATM networks [RMV95], we can write the following expression for the recalculated value of token bucket size, \( \hat{b}^\prime \):

\[
\hat{b}^\prime = b + \hat{IPDV} \cdot r
\]

(4)

where \( b \) is the original declared bucket size and \( r \) is the original token bucket rate. Notice, that this expression constitutes a part of the specification of the delay function block for the particular domain (figure 3.3).

3.1.2.2. Numerical results

The effectiveness of discussed IDAC approach was analyzed by simulations using NS-2 [NS2] in a simple network consisting of 2 domains, named Domain 1 and 2, as depicted on figure 3.7.

![Figure 3.7: The studied system.](image)

The flow considered for IDAC will be established between the source host (S) and the destination host (D) belonging to different domains. We assume that both domains offer an exemplary QoS network service, designed for handling variable bit rate traffic with negligible losses. The traffic submitted into this service is described by double token bucket with the following parameters: peak rate \( P \), sustained rate \( r \) and corresponding bucket size \( b \). For such defined service, the admission decisions inside each domain may be performed based on the following formula for effective bandwidth [RMV95]:
\[
\text{eff}(\cdot) = \max \left( \frac{P}{1 + B \cdot \left( C \cdot b \right) \left( P - r \right)} \right) 
\]  
(5)

where \( P, \ r \) and \( b \) are TD parameters, while \( C \) and \( B \) are the link capacity and associated buffer size, respectively.

In the IDAC approach, as it was presented in section 3.1.1, the admission to domain 1 is performed based on TD\textsubscript{1} with \((r,b)\) originally declared by a user, while in domain 2, based on re-calculated TD\textsubscript{2} \((r,\hat{b})\), taking into account the assessment of traffic profile deformation introduced inside domain 1. Below, we focus on evaluation of the traffic deformation introduced inside the domain 1 and then we show its impact on performing AC in the domain 2.

3.1.2.2.1. Evaluation of traffic profile deformation

In this section, we study the effectiveness of the proposed TD recalculation method. For that purpose, we assume that Domain 1 consists of a chain of \( m-1 \) routers, \( N_{i}, \ i=1,\ldots,m-1 \) connected by links with capacities \( C=2\text{Mbit/s} \). Within each router, the analyzed foreground flow is multiplexed with the background flow, \( BG_{i}, \ i=1,\ldots,m \), injected independently on each link, as depicted on figure 3.7. Assuming large degree of possible aggregation of individual flows, the background traffic can be modeled as a Poisson process.

Figure 3.8 shows how the traffic profile deformation accumulates with the number of routers in a chain. In this case we assume that foreground traffic is deterministic ON/OFF flow with \( P=200\text{kbit/s}, \ r=100\text{kbit/s}, \text{MTU}=500\text{B} \) and the corresponding value of \( b \) was fixed between 1 and 100 MTUs, which reflects different length of ON and OFF periods. The background traffic load on each link was fixed at \( \rho=0.8 \).

The traffic profile deformation is expressed as the value of \( \hat{b} \) in relation to \( b \), where \( b \) is the parameter of the original token bucket TD and \( \hat{b} \) was obtained by simulations as the minimum size of the token bucket applied at the egress point of Domain 1, such that the probability of a non-confirming packet was negligible. Thus, value of \( \hat{b} \) represents the actually required update of original TD parameter, \( b \). On the other hand the value of \( \hat{\hat{b}} \) was obtained by formula (4) from measurement of \( \hat{\text{IPTD}} \) estimated taking into account IPTD of all packets belonging to the foreground flow. Notice, that \( \hat{\hat{b}} \) constitutes a conservative upper bound for the true required \( b' \) in all evaluated traffic scenarios. The level of overestimation depends on the characteristics of the particular traffic source.
The effect of traffic deformation is especially visible in the case of flows characterized by rather small values of $b$, the original bucket size. For flows with large values of $b$, the deformation has a maximum in certain value of $m$ and then it diminishes. The widths of 95% confidence intervals are indicated on all plots.

![Figure 3.8: Difference between $b'$ and $b$ vs. the number of routers in a chain, $m$.](image)

The next experiment was carried out in a network with $m=10$ routers and with foreground flows characterized by different values of peak rate, $P$. The mean rate was always $r=P/2$ and the durations of ON and OFF periods were fixed in such way, that the corresponding value of $b$ was equal to 5 MTU. Simulations were repeated under different background traffic load conditions, represented by the value of utilization coefficient, $\rho$. On figure 3.9, the value of obtained $b'$ normalized in relation to the original bucket size $b$, is depicted as a function of ratio of $P$ to $C$. Notice that the traffic profile deformation effect increases with the relative “size” of the flow with respect to the total link capacity.
Based on the obtained results one can conclude that the traffic submitted by a user changes after passing the domain even significantly and that the original TD is inaccurate. Therefore, it has to be recalculated and the discussed measurement-based method can be effectively applied for that purpose. However, the proposed method overestimates the TD.

3.1.2.2.2. Evaluation of IDAC approach

Now, we investigate the impact of deformation of traffic profile in Domain 1 on the effectiveness of AC performed within the Domain 2 and, as a consequence, on end-to-end QoS offered to the user. For that purpose we analyze the impact of TD changes on the value of effective bandwidth calculated according to formula (5). In figure 3.10 we show the value of effective bandwidth, $e'$, calculated based on the modified TD obtained in the experiments described in section 4.1, in relation to $e$, the effective bandwidth calculated using the original TD.
One can observe, that the level of TD deformation will have impact on the number of admitted flows. In order to practically verify that performing inter-domain AC without taking into account the effect of traffic deformation may lead to QoS failures, we performed the following simulation experiment. Let us assume, that the capacity of the bottleneck link L in Domain 2 (see figure 3.7) is $C=2$Mbit/s, while the size of the associated buffer, $B=22000$Bytes. Further, assume that a flow (denoted as flow type #1) has TD parameters: $r=100kbit/s$ and $B=22000$bits. According to formula (5), the maximum admissible number of type #1 flows on link L is 14.

Simulation experiment, carried out with 14 simultaneous flows, generated between nodes $N_1$ and $N_2$ exactly according to the ON/OFF profile determined by TD of flow type #1, confirms that the packet loss ratio measured on link L is equal to 0 (see table 3.1, row 1).

Then, assume that one of the flows passing link L has arrived from the Domain 1. Its traffic profile would be deformed, and the correct TD value for this flow is $b'=25670$bits (obtained in simulation scenario 1). In the experiment, one of the flows on link L was exchanged with flow type #2, generated according to profile determined by the modified TD with parameters $(r,b')$. Now (see table 3.1, row 2), one can observe packet losses on link L. This suggests, that the effective bandwidth of flow #2 submitted to link L should be calculated not based on original TD $(r,b)$, but based on recalculated TD $(r,b')$. The admissible set would then be reduced to 12 flows of type #1 and 1 flow of type #2, but target packet loss ratio equal to 0 is kept (see table 3.1, row 3). The result presented in the last row of table 3.1 was obtained with the effective bandwidth of flow #2 calculated based on the TD estimated with the help of
measurements, represented by \((r, \hat{b}')\). The admissible set is here also equal to 12 flows of type #1 and 1 flow of #2.

<table>
<thead>
<tr>
<th>TD of flow #2</th>
<th>Flow #2 TD bucket size</th>
<th>Flow #2 real bucket size</th>
<th>N. of flows #1</th>
<th>N. of flows #2</th>
<th>P(_{\text{loss}})</th>
</tr>
</thead>
<tbody>
<tr>
<td>((r,b))</td>
<td>22000</td>
<td>22000</td>
<td>14</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>((r,b))</td>
<td>22000</td>
<td>25670</td>
<td>13</td>
<td>1</td>
<td>3.8 (\times) 10(^{-2})</td>
</tr>
<tr>
<td>((r,b'))</td>
<td>25670</td>
<td>25670</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>((r,\hat{b}'))</td>
<td>30400</td>
<td>25670</td>
<td>12</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 3.1: Packet loss ratio experienced by flows submitted to link L, assuming unmodified and recalculated TDs of flow #2.

The results of this simulations confirm, that recalculation of TD is desired for the IDAC in order to keep stringent QoS guarantees. However, the exhaustive evaluation of impact of traffic deformation needs more experiments with different traffic loads and real IP QoS network topologies.

3.1.3. **Summary and open issues**

We discussed some of the issues related with performing admission control in a multi-domain IP QoS network. For that purpose, the general model of multi-domain network was introduced. This model constitutes a base for designing and performance analysis of inter-domain admission control.

Some open issues related with performing IDAC have been identified:

- **Problem of traffic characterization**
  
  We discussed one of the possible solutions to the problem of traffic profile deformation. The proposed admission control approach is enhanced by a method for recalculation of traffic descriptors. The motivation for performing recalculation is that after passing a domain, the traffic may have worse properties than the originally declared profile. The simulations confirm, that the effect of traffic profile deformation can be quite important and neglecting it may lead to degradation of end-to-end QoS. This is especially critical for the tight AC algorithms, which give the exact boundary and even small inaccuracies in TD lead to QoS degradation.

  Notice, that the proposed measurement-based method of TD re-calculation is only one of the possible approaches for assessment of traffic deformation. Other approaches giving lower over-dimensioning of TD should be considered. The impact of the effect of traffic...
profile deformation in the case of other network services should also be carefully investigated.

- Inter-working (mapping) between services offered in different domains
  In the general case, QoS services offered by different domains do not have to be identical. The differences can be related with the level of offered QoS guarantees, the methods for traffic characterization, traffic handling and admission control. In order to provide consistent end-to-end QoS service, the rules for possible inter-working between these different services must be precisely defined.

- QoS splitting/assembling functions
  The flow QoS requirements are expressed in terms of end-to-end parameters, thus they are not directly related with the QoS level offered within a single domain. Therefore, on each step on the end-to-end path the IDAC has to assess the QoS degradation level expected in particular domain and inform the next domain that it has to offer e.g. lower delay and packet loss ratio, taking into account the degradation introduced in all previously visited domains. This task may be realized with the help of so-called QoS assembling functions, which compose the end-to-end QoS values based on the contributions of consecutive domains. This is also a quite complex problem due to the variety of QoS parameters and their features. For example, the maximum packet delay is additive while the packet loss ratio is rather multiplicative.

3.2. Admission Control for SLA Management

Two are the approaches that have gained a significant consensus for the provision of QoS guarantees over the Internet: IntServ and DiffServ. While IntServ is based on explicit signalling for the provision of QoS guarantees on a per-flow basis, DiffServ requires packet classification at the network ingress, and provides a differentiation of the treatment according to a (small) set of classes, named Per-Hop Behavior (PHB). In the DiffServ framework, a service contract, or Service Level Agreement (SLA), is established between a customer and a service provider, to specify the forwarding service that a customer should receive. The various PHBs define a rich toolbox for differential packet handling by individual IP routers. Common PHBs are Expedited Forwarding (EF), and Assured Forwarding (AF), which includes the classic Best Effort (BE). The EF PHB appears to the endpoints like a point-to-point connection, or a “virtual leased line”, on which a deterministic QoS guarantee is offered. The AF PHB instead offers a soft QoS guarantee: within each AF class, IP packets
are marked with one of three possible drop precedence values. If congestion arises in the network, a congested node tries to protect packets with better service profile by preferably discarding packets with a higher drop precedence value. The traffic offered by the user is metered at its ingress node according to the user's traffic profile, and packets are marked with the contracted drop precedence value if the data rate is below the contracted rate; otherwise, they are marked with a higher drop precedence value. The AF PHB can be used in a point-to-point setting, as well as in point-to-multipoint configurations, where traffic can flow to different destinations (or come from different sources) at the same time. 

When considering SLA admission control in a Differentiated Service (DiffServ) domain, the EF class can be considered a classic approach to provide QoS guarantees, and therefore classic admission control scheme can be applied to traffic belonging to this class. More interesting is instead the AF class, which from one side provides soft guarantees to traffic and from the other side offer a more flexible traffic definition.

The main conceptual difference between classic admission control problems and the AF-SLA admission control problem in a DiffServ network consists in the SLA definition itself. Indeed, in a AF-SLA, the SLA contracted between the network and the user involves in general more than one source/destination pairs. In particular, two possible types of SLA are possible: the first refers to traffic generated by a single user and directed to (possibly) multiple destinations, while the second considers traffic going from (possibly) multiple sources to a single user. Each SLA is described in terms of parameters, such as assured bandwidth going to (coming from) a set of possible destinations (sources) within the same DiffServ Domain, or, possibly, egress (ingress) nodes connected to other domains.

The choice of the links which will then be used to carry the traffic belonging to a given AF-SLA is therefore different from the classic routing problem in a point-to-point requests. It indeed can be formalized as finding the set of paths which will be used to route a new SLA request, so that the QoS experienced by all AF-SLAs over such set of paths is guaranteed.

This problem is equivalent to the well-known Steiner Tree problem, which can be summarized as the problem of finding the minimum cost tree that connects a source node to a subset of vertexes in digraph. Several cost functions can be considered, most of which can be derived from classic admission control problems. Being the Steiner Tree problem NP-complete, it can be solved only using heuristics.

The problem of finding the best Steiner tree has also been tackled in the optimization of Virtual Private Networks, and the literature traditionally refers to it as the “hose model”.

---

49
A VPN is an emulation of services provided by Private Networks (PN) - leased lines connecting a set of sites. With an increasing number of VPN endpoints, an alternative to the standard point-to-point (or customer-pipe) approach has been proposed in the hose model [DUFF99]: given a network, whose link capacities are known, and a set of VPN sources/destinations, one needs to know an upper limit of the amount of traffic entering and leaving each node from and to the other nodes. Then, a VPN hose is created connecting all nodes, with preference for sharing as many links as possible. The structure of the hose can be a tree, or any other connected graph. Once the set of routes (or the Steiner Tree, or the hose) has been defined, classic approaches to test if the network has enough resource to allocate to the novel SLA request can be applied. If not enough resources are available, then the SLA will be rejected.

In the literature, SLA admission control is addressed by both Measurement-based (MB) approaches and Parameter-based (PB) approaches. End Point Admission Control Through Probing [Breslau00] is the most interesting MB approach. Probes are sent out in the requested traffic class and the results gathered by the egress router in terms of packet loss, jitter, transmission time etc. The egress router itself decides whether the flow can be accepted. This has the problem of bandwidth stealing, since the probes do not show the effect of accepting the new traffic on already accepted SLAs. Also there is an overhead in the form of large setup time.

More traditional approaches include pricing mechanisms, which are proposed as a viable means to the SLA admission control. One such proposal can be found in [WANG01], for example, whose authors propose a congestion-sensitive pricing scheme based on the cost of providing different levels of QoS to different DiffServ classes, and on long-term demands. They also develop a demand behavior of adaptive users based on a user utility function. Explicit admission control is avoided thanks to the congestion-sensitive pricing policy: users lower their requests (i.e., choose a different DiffServ class of service) in the face of increased costs.

Other approaches are hinged upon backward learning algorithms, such as the one proposed in [HUI03]: an adaptive provisioning mechanism based on reinforcement learning principles determines at regular intervals the amount of bandwidth to provision to each PHB behavior. The mechanism adjusts to maximize the amount of revenue earned from a usage-based pricing model. Accurate traffic specification and rigid admission control are avoided by using a continuous-space gradient-based learning algorithm.
Other schemes follow more standard approaches. For example, [OOTT01] couples a MB admission control for SLAs with deterministic envelopes to characterize arriving traffic and service. In [RHEE02] bandwidth brokers are introduced to centralize admission control using a distributed measurement based admission control.

3.3. **SLA/SLS Management in Multi-domain Environment**

3.3.1. **Introduction**

The Internet evolves toward the global multiservice network, including support for services with guaranteed end-to-end (E2E) quality of service (QoS). Several technologies have been proposed. The *DiffServ* [RFC2474], [RFC2475], [RFC2597], acts in the control/data plane mechanisms to support QoS, but is not end-to-end and cannot assure precise hard guarantees. *IntServ*, [RFC2210], [RFC2211], [RFC2212] uses a per flow signalling, being an alternative which can assure E2E QoS but is not scaleable. *IntServ* in the edges and *DiffServ* in the core have been proposed, [RFC2298] but still the problems of E2E persist as not totally solved, especially in large inter-domain environment. In fact the DiffServ has been proposed to provide QoS in a scaleable fashion. Instead of maintaining per-flow soft state at each router, packets are classified, marked, and flows are treated conforming to their relative priority, but the technology does not assure itself E2E QoS delivery. Therefore the E2E QoS problem is still open especially in multi-domain environment. The QoS characteristics and the levels of guarantees for the end user are established and controlled by some contracts between end users and providers or between providers.

This section is a short presentation on the state of the art for QoS based services related to the contracts named *Service Level Agreement (SLA)/ Service Level Specification (SLS)* between the customer and provider of services (ISP), and their management. It links this problem to the *Policy Based Network Management (PBNM)*, model [RFC2753], [RFC2748], and analyses the architectural framework to support SLA management within PBNM framework, for DiffServ, with final goal to assure E2E QoS in a heterogeneous environment. First the business models are introduced defining the actors involved. The PBNM basic concepts are recalled. The SLA definitions and framework is introduced. The intra-domain and inter-domain treatment of SLA/SLSs are discussed with emphasis on two-time scales approach for both service management & control and respectively of resource & traffic management and control.
3.3.2. **Business Models**

Several business models are assumed in different approaches and QoS related projects and experiments. A basic set can be listed below.

*User/Customer* (can be a company, residential user, application/content provider, another provider), *Reseller*.

The providers are: *Service provider* (SP), *IP Network provider/operators* (NP/NO), *Physical Connectivity Provider* (PHYP). The NP can play in some cases also the role of PHYP.

Usually the NP/NO have one or more Autonomous Systems (AS) to manage. Other entities than those mentioned above can appear in more refined business models: *Access Mediator* (AM), [Cadenus], [Cort03], present between end user and the service provider (SP), or **Service Aggregator** - an entity which deliver to its customers a set of several services collected from several SPs. All these types of providers will cooperate in providing integrated and widespread services, while maintaining their business identity. Occasionally, a single service provider can cover more than one activity area for a certain business. They will have to disclosure inside information regarding their performances and resources availability but also to hide them from competitors. Different SLAs and NDAs will be enforced between them, but the user will agree on only one integrating SLA. In the text below we will only consider the basic set of business actors.

The portfolio of services will include a lot of voice, audio, video and multimedia, internet access, etc., fixed or mobile services. For each service a business model is to be developed and customised by any service provider.

3.3.3. **Policy Based Network Management Concepts**

The PBNM is a powerful set of concepts and methods able to make the network management more powerful, flexible and dynamic. PBNM is a methodology specifying behavior parameters in an abstract way. In a heterogeneous system, built with several equipments from different manufacturers, those policies can result on different configuration settings according to the characteristics of each managed device. One application suggested is the QoS control in IP networks. Here we shortly recall the main definitions, used in this framework, [RFC2753], [Stev99], [Cort03].

The *Policy* is a combination of rules and services where rules define the criteria for resource access and usage. The *Policy control* represents the application of rules to determine whether or not access to a particular resource should be granted. A *Policy Object* contains policy-related information such as policy elements and is carried in a request or response related to a...
resource allocation decision. The Policy Element is a subdivision of policy objects; it contains single units of information necessary for the evaluation of policy rules. A single policy element may carry an user or application identification whereas another policy element may carry user credentials or credit card information. The policy elements themselves are expected to be independent of which QoS signaling protocol is used.

The Policy Decision Point (PDP) is central element in PBNM, representing the point where policy decisions are made. The execution of policies are performed by one or several Policy Enforcement Points (PEP) which are the points where the policy decisions are actually enforced. In real network there can exist some Policy Ignorant Nodes (PIN) which are network elements that do not explicitly support policy control by using the mechanisms defined in PBNM context.

The network has Resources that are something of value in a network infrastructure to which rules or policy criteria are first applied before access is granted (e.g. buffers in a router, bandwidth on an interface, etc.)

The PBNM suppose to use a Soft State Model. A Soft state is a form of the stateful model that times out installed state at a PEP or PDP. It is an automatic way to erase state in the presence of communication or network element failures. (e.g. RSVP uses the soft state model). An Installed State is a new and unique request made from a PEP to a PDP that must be explicitly deleted. A Trusted Node is a node that is within the boundaries of an Administrative Domain (AD) and is trusted in the sense that the admission control requests from such a node do not necessarily need a PDP decision.

Each domain may contain one or more Policy Servers (PS) to make policy and configuration decisions for network elements. The PS has access to a policy database (e.g. LDAP or SQL) as well as authorization and accounting databases. Each policy entry specifies a rule of “if condition happens, then take certain action”. A human operator uses a GUI management application which interfaces to the PS through a set of Policy API (PAPI). This allows updating and monitoring policy changes in the policy database. The PS consists of a Central Policy Controller (CPC) and a set of PDPs. The PDP’s determine which actions are applicable to which packets. The CPC is to ensure global consistency between decisions made by the PDP’s. The enforcement and execution of policy actions are done by PEPs. PEP’s are typically colocated with packet forward components, such as border routers. PDP’s can interact with PEP’s via Common Open Policy Service (COPS) protocol. PDP’s push configuration information down to the PEP’s as well as respond to queries from the PEP’s.
The Bandwidth Broker (BB) concept is used to implement PDPs, [RFC2638]. BB is a logical resource management entity that allocates intra-domain resources and arranges inter-domain agreements. A BB of each domain can be configured with organizational policies and controls the operations of edge routers. In the view of policy framework, a BB includes the function of PDP and policy database, while edge routers serve as PEPs. In its inter-domain role, a BB negotiates with its neighbor domains (in so called “cascade model”), sets up SLAs with each of them, and sends the appropriate configuration parameters to the domain’s edge routers. Alternatively a BB can discuss with a central entity in so called “hub model”. Bilateral agreement means that a BB only needs to coordinate with its adjacent domains. E2E QoS is provided by the concatenation of these SLSs across domains, together with adequate intra-domain resource allocation. Within a domain, a BB performs resource allocation through admission control (AC). The choice of the intra-domain algorithm is independent of the inter-domain negotiation.

3.3.4. Service Definitions, Service Level Agreements

From the customer view a Service is a specific offering made by a provider, clearly and unambiguously describing what it offers and the terms and conditions under which it could be used. From the provider view a Service is a subset of the provider's domain capabilities with a clear description of the what is and how is regarding its use by customers or third parties in general. In IP networks a QoS based service is the QoS-based connectivity service - a traversal service for reaching particular destination(s) from specific source(s) in the IP address space. The QoS aspects of connectivity services mainly refer to the quality at which the user-transmitted IP datagrams are transferred by the network between user-ends. The QoS levels and guarantees are specified in the SLA contracts.

The Service Level Agreement (SLA) is a documented result of a negotiation between a customer and a provider of a service that specifies the levels of availability, serviceability, performance, operation or other attributes of the transport service. A provider may also have the role of a customer, i.e. asking for transport connectivity service to another service provider. The SLA contains technical and non-technical terms and conditions. The technical specifications, (IP QoS guarantees and traffic parameters) are part of the SLA.

The Service Level Specification (SLS) is a set of technical parameters and their values, defining the service, offered to a traffic stream by a DiffServ domain. To provide value-added IP services, the DiffServ network should be able to provide some (predictable) edge-to-edge QoS guarantees. This edge-to-edge QoS behaviour is described by QoS-Class. The SLS
content can be, [God00], [Sals00]: Scope (1-1, 1-n, n-1); Flow descriptor (flow identification: source, destination addresses, DSCP); Traffic descriptor; Excess treatment (re-marking, dropping); Performance parameters (delay, throughput, loss, delay jitter); Service time schedule; Reliability.

The SLA Templates are used during Customer negotiation to define the required level of service quality. The production of an SLA template is an intrinsic part of service development. These SLA templates may relate to “standard” product/service offerings, where they are used “as-is” to define the required level of service quality or as a baseline for custom negotiation (either automated or human-assisted). SLAs are defined on something perceived by the Customer (i.e. explicitly subscribed to), that is the service elements composing the service product offering.

The SLA/SLS are uni-directional. For bi-directional case two contracts should be established. In multi-domain environment one can define two types of SLS, [Tequila], [Mescal], [Trimin01], in line with the classification of entities, which set out the service levels between the involved entities, based upon a set of QoS parameters.

\[ pSLS \ (p = \text{“Provider”}) - \text{SLSs between NP and SP ASs; cSLS } (c = \text{“Customer”}) - \text{SLSs between CSs and the AS of the Service Provider chosen for the request of certain media/data flow. Between operators/domains the pSLA/SLSs can be established on aggregated basis.} \]

The subscription of a SLA implies two aspects: the auditing of the actual satisfaction of current SLA and the dynamic re-negotiation of the SLA: when the users start to pay for communication services with QoS-related guarantees it will be required to verify whether or not the provider actually fulfills the SLA. Re-negotiation of QoS is an important option in performance guaranteed communications. The re-negotiation of an SLA can be triggered by several events linked to critical problems such as the efficiency of network resource allocation, the performance of user’s application, and the reduction of communication costs.

3.3.5. QoS classes of services
A QoS-class (QC) denotes, [Tequila], [Mescal], (shortly) a basic network-wide QoS transfer capability of a Provider domain. The QC provides certain edge-to-edge QoS guarantees, (packet loss, delay, throughput) to a traffic stream and the packets belonging to these streams. A QC is not defined within the context of the DiffServ Workgroup. The QoS guarantees implied by Per Domain Behaviours (PDB) or QoS-Classes will be provided by the support of the dedicated PHBs in the (core) routers and the Traffic Control (TC) functional blocks at the
edges. The actual implementation of the PHBs and TCs in the routers is done through particular schedulers, queuing algorithms such as RED, shapers, meters, etc.

A QC can be specified as: OA (ordered aggregate), delay-bound, loss-bound, [jitter related bounds] where OA denotes the PHB Group with which the packets of this class will be treated (it may take the values: EF, AF1, AF2, AF3, AF4).

QoS is completely specified in terms of the QoS-classes. One has to distinguish between the QoS-aware services and QC. The Services denote the IP connection capabilities (expressed in network technology-independent semantics) that can be offered. The QCs denote the elementary transport capabilities that can be gracefully supported by the network through the employed Traffic Engineering (TE) functions.

Different models of QCs can be used. A simple model, [Engel03], can define a small set of well known classes dedicated to certain applications like: PCBR (Premium Constant Bit Rate), PVBR (Premium Variable Bit Rate- VBR ), PMM (Premium Multi-Media), for greedy and adaptive applications, PMC (Premium Mission Critical), for non-greedy applications. More sophisticated usage of classes are defined in [Tequila], [Mescal]. In multi-provider environment composition rules to chain the local QoS classes should be defined in order to get an E2E SLA.

3.3.6. QoS guarantees

The levels of QoS guarantees depends on the category of end-users and on the price. Also they will have different requirements for the topological scope of their SLS and will behave differently from the point of view of invocation of services. The end-users can be can be:

- **Residential** customers which may subscribe to delivered services. They may want to reach any or a limited set of available destinations at any time or in some time windows defined in SLA. The frequency of service invocations and the transaction duration can be very variable depending on the service. This type of users could be satisfied with a best/good/moderate quality of the service depending on the price.

- **Organisational/Corporate** customers which may want specific, strong guarantees for supporting particular mission- or safety-critical applications and services. Usually the destination set is limited and the frequency of transactions could be high. These customers will probably accept to pay the higher price of some hard guarantees of QoS.

A range of guarantees levels can be, [Mescal] and associated service options are:
• **Loose Guarantees**: globally aims at providing better than best effort based services, but doesn't provide any strong guarantees (qualitative E2E QoS performance: delay, jitter, loss; no bandwidth guarantee, any reachable destination is allowed).

• **Statistical Guarantees**: statistical QoS performance guarantees for specific destinations, some loose end-to-end bandwidth guarantees, while restricted to some specific destinations.

• **Hard Guarantees**: quantitative end-to-end QoS guarantees, bandwidth guarantee, specific destinations.

The problem of co-existence of several levels of guarantees in one AS should be solved. In the inter-domain case an additional problem appears if the chaining ASs offer different types of guarantees.

3.3.7. **Intra-domain aspects**

3.3.7.1. **Architectures**

The architectural planes allowing to apply E2E QoS technologies and PBNM concepts are usually: *Service Plane, Management Plane, Control Plane and Data Plane*. In case of multi-provider context the first three planes should functionally cover the intra-domain but as well the inter-domain service management, resource management and control and traffic engineering.

3.3.7.2. **Service (SLA/SLS) subscription and invocation**

The E2E QoS architecture architectures should facilitate the functional decomposition of the main service and resource management aspects.

One key concept, [Mykon03], is to adopt a *two-level approach* for (operational) service management and negotiation, i.e. service subscription (SLA/SLS setup) and service invocation. Both processes occur at a different time scale. Subscription (processed by the Service Plane) handles the longer term-based service requests that may apply to IP services. The SLA/SLS are only promises about some resources. The actual allocation can happen later on at service invocation epochs. The service invocation (processed by the Control plane) can happen immediately after subscriptions or later. At the edge of the network the invocation acts on a per-call basis. Between the domains the invocation means in fact pSLS invocation, followed by real allocation of resources in each AS composing the chain of SLA/SLSs. The two-level approach in service management is mirrored in the resource management system. The architecture combines a longer term off-line traffic engineering approach (*network
A second concept is the distinction between the customer (SLS) aware components and the resource (QoS class) aware components. The service manager knows about the customers, while the resource manager knows about the network resources, and acts on the processing of (aggregate) traffic that will be handled by a collection of QoS classes. The inter-working between the two sub-systems can be defined through the so called resource provisioning cycle, [Trimin01], [Mescal], controlling the interactions between several elementary components: service subscription, traffic forecast and network dimensioning.

In fact this approach proposes a two-level approach of admission control, mirrored in the Two time scale level approach for resource management: managing long term IP aggregates based on a Resource Provisioning Cycle (RPC); short term flows - invocations with guidelines set by RPC.

A hierarchical service model can be adopted which translates from (SLAs) down to per hop behaviors (PHBs) in the network nodes. The QoS classes depict the elementary QoS transfer capabilities of an SP domain, consisting of an ordered aggregate (OA) and associated QoS parameters such as one-way delay and packet loss. Each service corresponds to a number of SLSs, and each SLS corresponds to a number of QoS classes. Therefore, given a service, its QoS is completely defined through the QoS classes of its constituent SLSs. The following top-down mapping starting from high level SLA down to network elements can be used [Tequila], for DiffServ Layered Service Model:

- **SLA Transport Service** (non-technical terms & conditions technical parameters -{SLS}-set)
- **SLS QoS class** (IP service traffic characteristics, offered network QoS guarantees)
- **Per Domain Behaviour (PDB)** (network QoS capabilities, DiffServ edge-to-edge aggregates)
- **Per Hop Behaviour (PHB)** (generic router QoS capabilities, DiffServ core & edge routers)

**Data plane mechanisms:** Traffic Conditioning Block, Scheduler (e.g. WFQ), Dropper (e.g. RED)

### 3.3.7.3. Resource and Traffic Management

The resource and traffic management is not the main focus of section 3.3. We only present some connections of this topic with the SLA/SLS management.

Corresponding to the two level of SLA/SLS treatment, one can define a Long term network resource management and traffic engineering and respectively a Dynamic Resource and Traffic Engineering Control, [Mykon03], [Tequila]. A long term network provisioning cycle
can be defined comprising the following functions and data: Subscription Data (SLA/SLS) → Traffic Forecast Algorithm → Traffic Matrix Data → Long term TE (network dimensioning) → Resource Availability Matrix (RAM) Data → Long term admission control (of new SLA/SLS) → Subscription Data.

The dynamic control will consist of Dynamic Route and Resource Management and Service Invocation Admission Control. Dynamic Route Management (DRtM) manages the routing processes in the domain according to the guidelines produced by Network Dimensioning (ND) on routing traffic according to QoS requirements associated to such traffic, i.e. the contracted SLSs. Dynamic Resource Management has an instance attached to each router. It ensures that link capacity is appropriately distributed between the PHBs sharing the link. The Service Invocation Admission Control scheme can be based on the resource availability estimates calculated by the offline TE functions in the RAM per traffic trunk (TT) monitored data reflecting the current status of domain resource utilization.

### 3.3.7.4. Specific protocols

The generic protocols necessary in intra-domain QoS aware network are: QoS aware intra-domain routing protocol, (e.g OSPF like) SLA/SLS negotiation protocol and Resource Allocation protocol used at SLA/SLS invocation epochs. These include the signalling between the Service/network provider and user terminals.

### 3.3.8. Inter-domain aspects

#### 3.3.8.1. QoS peering methods

In multi-provider environment, the SLA/SLS should be established between the domains (ASs/NPs, SPs). Several models of ASs peering exist, [Mescal], in order to ensure an E2E QoS based service. The type of inter-domain peering impacts the service negotiation procedures, the required signaling protocols, the QoS class mapping/binding and path selection. The hub model supposes that the Service Provider (SP), is a distinct entity from NP. The SP is a central point that negotiates and establishes provider-level SLSs with all individual NPs to be involved in the service chain. The centralised model, is similar to the hub model, where the first Network Provider (NP) in the chain selects one or several downstream providers and negotiates provider-level SLSs directly with them, in order to construct an end-to-end QoS service. The cascaded/bilateral model supposes that each NP only negotiates provider-level SLSs with its immediate neighbouring provider(s) to construct an E2E QoS based service. Inter-domain routing information are required to indicate the route.
towards a given destination. That is, only neighbouring domains establish provider-level SLSs between themselves. The cascade model can perform then set up of a chain SLA/SLS in forward mode (the same direction as data flow) or in reverse mode. The cascade model is more scalable for large multi-provider environments because it avoids that a central point should exist which must know the whole topology in order to be able to determine the inter-domain route between two or several points.

### 3.3.8.2. Architectures and protocols

The architectural planes allowing to develop E2E QoS technologies and PBNM concepts are the same as in intra-domain case: Service Plane, Management Plane, Control Plane and Data Plane. In case of multi-provider context, the first three planes functionally cover the intra-domain but as well the inter-domain service, resource management and control and traffic engineering. If a BB based architecture is used, then an intra-domain and inter-domain part of BB should be considered.

The generic inter-domain protocols necessary in multi-provider QoS aware network are:

- QoS aware inter-domain routing protocol, (e.g BGP like, but enhanced for QoS constraints);
- pSLA/SLS and cSLA/SLS negotiation protocol used when setting up the SLA/SLSs contracts;
- Resource Allocation protocol used at SLA/SLS invocation epochs;
- Resource Monitoring protocol – to signal between network monitors of the domains.

These include the signalling between the Service/network provider and user terminals.

### 3.3.8.3. Service subscription and invocation

In inter-domain environment, the two types of SLA/SLSs are processed differently. Any pSLS is concluded (at subscription epoch) for a specific service class with identified and agreed characteristics and scope. The pSLSs are established between providers on aggregate basis. In a QoS-aware networking environment, only destinations that are within the scope of pSLS agreements in the chain are reachable. The pSLSs are always active because they are supporting aggregate traffic from many cSLS flows. A pSLS segment (between two ASs) can have as scope the ingress router of an AS up to the ingress router of the peer domain (in forward cascaded model).

The pSLS invocation is used when it is necessary to allow the user traffic to flow between the domains. The pSLS invocation may request (through a Resource Allocation protocol) a
certain amount of bandwidth (less than or equal to the value previously negotiated). Invocations usually happen at intervals less than a network provisioning cycle period. The pSLS bandwidth could be modified at request, in the interval between two runs of the network provisioning cycle if the network provider has available resources, but it should not disturb the other valid SLA/SLSs. A new resource provisioning cycle can be started asynchronously (at requests) if the number of c/pSLA/SLS refused is high, but this should not happen very frequently because it can disturb the existing SLA/SLSs.

The cSLSs are long-term contractual agreements or short-term agreements. Customers can dynamically subscribe and unsubscribe to QoS services, as per their communication needs. The cSLS may also need to be explicitly invoked. The scope of cSLSs are E2E. The traffic flows belonging to cSLSs, at the data transfer phase, will use the logical infrastructure that has been put in place based on the chain of pSLS agreements to get certain treatment. The cSLSs can be offered only after the pSLA/SLSs are established. The dynamic invocations are treated by AC blocks at ingress routers. During network operation, new cSLSs can be subscribed and invoked as long as their requested QoS requirements and bandwidth fall within the current respective pSLS agreements and there is no need for network reconfiguration.

**3.3.8.4. Resource and Traffic Management**

This subject is not the main topic of Section 3.3. The inter-domain Resource and Traffic Management deals with long-term actions concerning the inter-domain links and the traffic mapping on these links. This functional block will be in strong cooperation with the corresponding intra-domain entity.

**3.3.9. Examples of relevant architectures**

In order to illustrate the state of the art concerning the SLA/SLS management, we shortly present some approaches adopted in relevant IST FP5 and FP6 European projects applying the PBNM concepts for solving E2E QoS problems in single and multiple domain environment.

A number of concepts/features are common to all projects discussed which illustrate in fact what is the state of the art in the domain: they target to End-to-End QoS control with several classes of services and levels of guarantees; achieve QoS aware service in large network configurations; are based on PBNM concepts implemented in a distributed way; implement SLA/SLSs between customers and providers; use AC at service invocation epochs; use resource management and traffic engineering (TE) in order to optimise the utilization of
resources; use a monitoring system to provide feedback on network and resource status to the management function; are supported by DiffServ/MPLS technologies for traffic control.

The AQUILA project [Aquila], [Cort03], [Engel02], [Sals00], defines a limited set of network services in order to provide QoS differentiation at the service level in inter-domain environment. Each network service supports a class of applications with similar requirements. Service differentiation is applied at the packet level. Four traffic classes are defined beyond Best Effort (BE). For each traffic class there are defined: a scheduling/queuing mechanisms local to the router interfaces, the packet level conditioning functions at the network edge (marking, policing, shaping), and the call-level Admission Control (AC) criteria. All the AC and resource management mechanisms within the network are applied on a per-traffic-class basis.

The AQUILA is based on DiffServ architecture for large networks. It implements an overlaid distributed Resource Control Layer (RCL), as a mechanism for dynamic control of intra-domain resources, the Dynamic Resource Pool (DRP). In inter-domain it extends the Border Gateway Resource Reservation Protocol (BGRP) framework for the aggregation of inter-domain reservations to get scalability. Several interacting logical elements (RCL) can be seen as a distributed overlay control network on top of the DiffServ domain.

The RCL tasks are: interface to the QoS infrastructure for legacy applications through the End-User Application Toolkit (EAT) located in the end-user host; to do AC through AC Agent (ACA) located in each edge/border router; to monitor, control, and distribute the network resources through Resource Control Agent (RCA). The RCA interacts with the ACAs through the DRP mechanism. The RCA (node of RCL) is a logical unit that represents a portion of the IP network, which internally has the same QoS control mechanisms. An RCA is a generalisation of the BB concept. The QoS control mechanisms used in the network are admitted to be of varying nature, (e.g. DiffServ, etc.). The sub-networks are admitted domains managed by different operators.

The architecture enables the ACA to answer the AC question without interaction with a central instance, by using a mechanism of allocation of resources by the RCA to the ACA. The amount of resources allocated to ACA or returned by ACA can vary dynamically. The EAT is a middleware between the end-user applications and the AQUILA network infrastructure. It provides access to end-user applications to QoS features. The EAT supports QoS-aware but also Legacy Applications that are in fact QoS-unaware.
The network services are stored as XML data on a central directory server. A QoS Management Tool (QMTool) provides for network operators access to the network services. It retrieves the XML data from the server, modifies the entries in order to adapt the network service parameters, and finally to store the adapted entries on the server (new network services can be created in this way). Customers initiate via the End-user Application Toolkit (EAT) QoS requests by firstly selecting a network service, which can be seen as predefined SLS, and secondly by giving additional data for the chosen SLS. AQUILA uses the cascade model to achieve the SLS chaining.

The TEQUILA project, [Tequila], [God00], [Mykon03], [Trimin01], [Giord03], designs an architecture for intra-domain QoS delivery in IP DiffServ networks proposing service management and traffic engineering functions at various levels of timescale and abstraction. It uses QoS classes to link SLSs with Per Hop Behaviours (PHBs). QoS classes depict the elementary QoS transfer capabilities of an SP domain, consisting of an ordered aggregate (OA) and associated QoS parameters. Each service corresponds to a number of SLSs, and each SLS corresponds to a number of QoS classes. Given a service, its QoS is completely defined through the QoS classes of its constituent SLSs. The AC scheme was used based on a feedback-based model. AC logic is applied at both service subscription and invocation epochs. Feedback information is used at different levels of abstraction: input from offline TE functions on the ability of the engineered network to deliver QoS given subscription AC decisions, and input from measurements on the actual status of the network.

The architecture of a provider contains the sub-systems: Service Management (SM), Traffic Engineering (TE), Policy Management (PM), Monitoring and Data-plane. The data plane includes the DiffServ PHB and Traffic Conditioning Blocks (TCBs). The PM allows to define and enforce the policies for both SM and TE purposes in an automated way. Monitoring includes node monitoring, network monitoring and service monitoring. The SM includes service creation, negotiation and assurance. Service negotiation is the actual negotiation and subscription of value-added IP services between provider and customer. The Service Assurance (SA) enables the operator to verify whether the QoS performance guarantees committed in SLAs are being met in its network. The SA operates on the statistical data gathered by network monitoring through the network elements. TE specifies the manner in which traffic is treated within the network fulfilling both customer and system-oriented objectives. The expected performance depends on the type of traffic and is specified in the
SLSs. The target is to accommodate many QoS requests (as expressed in SLSs) by optimally using the available resources. The basic service implemented is IP connectivity services. TEQUILA has not in-depth studied the area of end-user services or service creation, i.e. the process of (automated) service definition by the provider. TEQUILA is not developing a complete (end-user) application service framework taking into account the business-related aspects and the different roles of the Internet stakeholders such as access providers, wholesalers, application provider, etc.

The SLA/SLS processing in TEQUILA is performed at the above mentioned two-level time scale. Subscription handles the longer term-based service requests that may apply to IP services (e.g. IP VPNs), while service invocation acts on a per-call basis (e.g. VoIP). TEQUILA focuses on operational management of IP connectivity services described by a set of SLS containing the description of the traffic profile and the network IP QoS guarantees, i.e. the traffic and resource related aspects of the IP service.

TEQUILA translates from SLA/SLS down to Data plane mechanisms as described in Section 3.3.7.2.

When a subscription request appears, its demand is aggregated with the demand of already established subscriptions. Total anticipated demand is estimated in terms of a minimum and maximum value corresponding to the almost and fully satisfied service rates of the subscribed services, respectively. A satisfaction level determines the portion of min and max demands to be safely accommodated in the network. Safe accommodation is deduced by checking whether the determined portions of minimum and maximum demand can fit in the area of the resource availability buffer, dedicated to this traffic even at congestion times. If they do fit, the subscription request is accepted; otherwise, is either rejected or alternatives of lower QoS and/or lower service rates are offered. The subscription logic can accept subscriptions either with no traffic considerations at all, on the grounds of their expected demand, or on a worst-case reasoning, depending on the particular setting of the satisfaction level (policy of the SP domain).

The MESCAL project [Mescal], extends the Tequila concepts to inter-domain IP environments. The domains cooperate to set-up QoS capabilities for building E2E QoS delivery chains. The QoS capabilities of each domain are supposed to be defined and having well known characteristics. Each provider in the E2E chain has to enforce its own intra-domain Traffic Engineering (TE) and routing protocols. MESCAL proposes several new
concepts on QoS classes (QC) and QC composition methods, different QoS inter-domain peering, for several levels of guarantees. No particular assumptions are made on what applications will use the offered QoS capabilities.

The MESCAL project is based also on distributed PBNM Architecture. The provider management plane contains the following functions:

Service Planning and QoS Capabilities Exchange (QoS based Service Planning and QoS capabilities discovery/advertisements – to define the QoS-based services and advertise to potential customers/service peers their characteristics;

SLS Management – to process the cSLS and pSLS requests undertaking authentication and authorisation tasks and ensuring that the network is not overwhelmed by traffic to the point that performance is deteriorated;

Monitoring and Assurance (SLS Assurance and Resource Monitoring) - provides raw data and derived statistics to the other entities and measures network performance to meet SLSs.

Traffic Engineering (Traffic Forecast and Off-line TE) – for configuring and controlling the necessary resources so that the services can be delivered within its domain and across service peer networks within the contracted performance levels. TE is separated in two functions: intra-domain and inter-domain.

The Control Plane includes SLS invocation and Dynamic Traffic Engineering.

SLS Management (SLS-M) handles the cSLS and pSLS - that specify the QoS connectivity services offered or acquired by the provider. MESCAL distinguishes into elementary (point-to-point and unidirectional) and complex connectivity services (multi-point-to-multi-point and bi-directional). Complex connectivity services encompass a number of elementary connectivity services as appropriate to the context of the connectivity service itself. SLS-M establishes agreements allowing the provider to expand its network’s QoS connectivity beyond its domain, and responds to the requests for services from the provider’s customers taking into account predictions of the network capacity and business directives. SLS-M can be split into two parts:

(a) SLS Order Handling: for the contracts offered by this provider to its customers, i.e. the end-customers and interconnected providers,

(b) SLS Ordering for the contracts requested by this provider from its peer providers.

The SrNP (Service Negotiation Protocol) is used for inter-communication between the two functional blocks.
Another separate process is required for service invocation to commit resources before traffic can be exchanged, with *SLS Invocation Handling* and *pSLS Invocation* providing the necessary functionality.

The “vertical” processing of SLA/SLS in MESCAL includes the following mapping (top-down):

\[
\text{SLAs} \rightarrow \text{cSLS/pSLS} \rightarrow \text{e-QC} \rightarrow \text{l-QC(PDB)} \rightarrow \text{PHB Groups} \rightarrow \text{Scheduling, Buffering mechanisms.}
\]

The “horizontal” processing of SLA/SLS consists in establishing SLS chaining to assure E2E services for a specified QC.

Three approaches are offered by MESCAL depending on the level of guarantees (*loose, statistical* or *hard*) required by the customer. All solutions assume that: DiffServ is deployed intra-domain but for *hard* guarantees MPLS is mandatory; intra-domain TE is deployed and cooperates with inter-domain TE.

*Loose guarantees:* the extended QCs (e-QC) relies on Meta-QC concept. The e-QC does not have to support bandwidth. The inter-domain signaling should transport information on Meta-QC used. A special protocol qBGP is used to find the paths within Meta-QC planes and optimize inter-domain routes. The qBGP selects and advertises one path for each Meta-QoS class per destination. Optionally an inter-domain advertisement on offered QoS classes (o-QC) could be used.

*Statistical guarantees:* the extended QC (e-QC) can rely optionally on Meta-QC concept. The e-QC have to support bandwidth constraints. The inter-domain signaling can optionally transport information on Meta-QC used. The qBGP protocol can be optionally used. Optionally an inter-domain advertisement on offered QoS classes (o-QC) can be used.

*Hard guarantees:* the extended QC (e-QC) can rely optionally on Meta-QC concept. The e-QC must support bandwidth constraints. The inter-domain signaling should transport information on Meta-QC used. The protocol qBGP is used to find the paths within Meta-QC planes. Optionally an inter-domain advertisement on offered QoS classes (o-QC) could be used.

*SLS Order Handling* perform subscription level AC. The *Off-line Intra-domain TrafficEngineering* block will provide *SLS Order Handling* with the resource availability matrix (RAM) – both within the AS and on any inter-domain pSLSs it has with neighbouring
ASs. **SLS Order Handling** negotiates the subscription of both cSLSs and pSLSs. **SLS Order Handling** maps incoming SLS requests onto the o-QCs it can offer and check whether there is sufficient intra- and inter-domain capacity, based on the RAM for that o-QC. Successfully negotiated SLSs are stored in the SLS repository (part of **SLS Order Handling**), and **SLS Invocation Handling** is configured appropriately to allow future invocations on the new SLS. The contents of the SLS repository are used as an input to **Traffic Forecast** for future RPCs. If there is insufficient capacity – either within the AS or on the pSLS with peer ASs – then the negotiation will fail. When RAM indicates low level then **SLS Order** can trigger a new RPC.

In MESCAL there exist pSLS invocation actions. The pSLS are always active (unlike many cSLSs which need to be explicitly invoked for each flow) because they support aggregate traffic from many cSLS flows, but the aggregate rate of the traffic may vary within RPCs. The **Dynamic pSLS Invocation** block monitors the usage of the pSLS, it can estimate future requirements and signal requests to increase or reduce pSLS capacity. Any increase or reduction in capacity should always be within the range allowed by the terms of the subscribed pSLS.

The CADENUS project, [Cadenus], [Giord03], is based on a distributed PBNM Architecture. It makes distinction/ separation between Service Provider and Network (Resource) provider. A special proposal of CADENUS is the **Mediation concept** involving the entities: **Access mediator** (AM - between end-user and service provider), **Service mediator** (SM- between AM and RM, creation and invocation of services), **Resource mediator** (RM between SM and the network management). The mediation procedure includes the mapping of user-requested QoS to the appropriate service-/network- resources, fit to business processes. AM presents the current service offers to the user. The source of the services is a **Service Directory** database. Generally off-line, the SM supervises the incorporation of new services, their presentation in the **Service Directory**, and the management of the physical access to these services via network, using the RMs. It is the task of the Service Mediator to prepare the SLA for the user to sign, and subsequently to authenticate the user and map the SLA from the AM into SLSs to be “signed” with the RMs. There is one RM per administrative domain (AS - Autonomous System) and one Network Controller for each network technology within that domain. RM is associated with the underlying network and its capabilities for supporting QoS, but the communication between SM – RM is independent of the technology employed by the network.
The end user contacts an Access Mediator (AM) in order to gain access to a number of services, by means of negotiation of specific SLAs. The AM interacts with one or more Service Mediators (SMs) (each providing a certain set of services), to retrieve information about the characteristics of the services and it organizes this information in order to allow the service selection. After selection, the involved SM(s) providing this service are notified and the negotiation process is initiated. The SM interacts with one or more Resource Managers (RMs) in order to negotiate the network resources needed. Eventually, the RMs configure the underlying network elements according to the QoS-related characteristics of the purchased service. A number of documents (SLA, SLS, policies) are generated in this process, each describing the same instance of the service at a different level of abstraction. These documents are created and interpreted by mediation components (AM, SM, RM) at the corresponding abstraction levels in the overall architecture.

The SM implements three processes: service design; resource negotiation with RMs; service fulfillment. The resource negotiation process accomplished by SM includes: finding partner RMs; setting up trading relationships with the RMs; acquiring information from the RMs; managing negotiations with the RMs. CADENUS applies a resource-based admission control at subscription and invocation time. If several network domains are involved, then an inter-RM communication is necessary to chain SLSs. CADENUS adopted the cascade model to chain the SLSs. CADENUS does not make a clear distinction between SLS established between the customer and the SM and a SLS established between two RM.

3.4. "Classical" vs. Chaotic-based Predictions

Accurate information on the future behavior of random processes is very relevant in many networking contexts (such as traffic control, data compression, resource allocation). Such a problem is typically known as “forecasting” or “prediction” problem and will be addressed in this section by presenting the application of some of the most common techniques based on classical and chaotic theories to network traffic.

The general problem of prediction can be stated as follows: given a set of observations of a stochastic process $x(n)$, estimate the value $x(n+k)$ that the process $x$ will assume $k$ steps ahead.
Such a problem is known as optimal $k$-step prediction, according to some optimality criteria. The predictor is called $k$-step optimal predictor. As an example, if the optimal criterion is to minimize the mean square error, the predictor will be denoted as MMSE (Minimum Mean Square Error) predictor.

Naturally, for the predictor to be practically useful, optimal criteria should obey a trade-off between estimation accuracy and a reasonable complexity.

Many approaches have been proposed in the literature to address the prediction problem. In this short presentation, we will present a selection of classical and “chaotic” prediction algorithms. In the framework of classical methods, we will focus on prediction techniques that do not require a prior detailed modelling of data to be forecasted. Chaotic predictors are presented in order to show a different philosophy to the prediction problem that, somewhat, takes into account the self-similarity of network traffic.

### 3.4.1 Classical Predictors

The following paragraphs are devoted to classical predictors. Out of the many that have been proposed in the literature, we present the ones that, because of their simplicity, happen to be more successful in the context of network traffic forecasting.

#### 3.4.1.1 LMMSE Predictor

One of the most widely used predictors is the so-called Linear Minimum Mean Square Error Predictor (LMMSE). Such a predictor is optimal in that, within the class of linear filters, it is the one that minimizes the mean square error of prediction.

Let us consider the stochastic sequence $x(n)$ (representing, as an example, the volume of traffic measured over the $n$th time unit) and suppose to be interested in the estimation of the value $\hat{x}(n+k)$, given a set of $p$ (possibly infinite) observations:

$$x(n) = [x(n) \quad x(n-1) \quad ... \quad x(n-p+1)].$$

The problem is to determine the impulse response $h(n)$ of the linear filter $h$ such that:
\[ \hat{x}(n + k) = x(n) \otimes h(n) = \sum_{i=0}^{p-1} h(i)x(n-i). \]  

(6)

According to the terminology of FIR filters, the predictor is called \textit{p-order LMMSE} predictor.

The \textit{prediction error} at the \(n\)-th step is defined as:

\[ \varepsilon(n) = x(n + k) - \hat{x}(n + k) = x(n + k) - \sum_{i=0}^{p-1} h(i)x(n-i). \]

(7)

The values \(h(n)\) are derived by minimizing the \textit{Mean Square Error} \(\sigma_{\varepsilon}^2 = E[\varepsilon^2(n)]\).

According to the \textit{principle of orthogonality}, the mean square error \(\sigma_{\varepsilon}^2\) is minimum provided the prediction error \(\varepsilon(n)\) has zero mean and it is orthogonal to each of the collected data. In other words:

\[
E[\varepsilon(n)] = 0 \\
E[\varepsilon(n)x(n-m)] = 0, \forall m = 0,1,\ldots, p-1
\]

Notice that, the last equation, can be re-written as:

\[
E \left[ (x(n+k)-\sum_{i=0}^{p-1} h(i)x(n-i))x(n-m) \right] = 0 \Rightarrow
\]

\[
\Rightarrow E[x(n+k)x(n-m)] = \sum_{i=0}^{p-1} h(i)E[x(n-i)x(n-m)]
\]

and finally, denoting as \(R_x(m) = E[x(n)x(n+m)]\) the autocorrelation function of the sequence \(x(n)\), the coefficients \(h(n)\) can be derived by solving the linear system (\textit{Wiener-Hopf equations})

\[ \mathbf{R_xh} = \mathbf{p}(k) \]

(8)

That, with obvious meaning of symbols, can be explicitly written as:

\[
\begin{bmatrix}
R_x(0) & R_x(-1) & \cdots & R_x(-p+1) \\
R_x(1) & R_x(0) & \cdots & R_x(-p+2) \\
\vdots & \vdots & \ddots & \vdots \\
R_x(p-1) & R_x(p-2) & \cdots & R_x(0)
\end{bmatrix}
\begin{bmatrix}
h(0) \\
h(1) \\
\vdots \\
h(p-1)
\end{bmatrix}
= 
\begin{bmatrix}
R_x(k) \\
R_x(k+1) \\
\vdots \\
R_x(k+p-1)
\end{bmatrix}
\]

The \textit{LMMSE predictor} is then obtained by inverting the previous relation, as:

\[
\begin{bmatrix}
h(0) \\
h(1) \\
\vdots \\
h(p-1)
\end{bmatrix}
= \mathbf{R_x}^{-1}
\begin{bmatrix}
R_x(0) & R_x(-1) & \cdots & R_x(-p+1) \\
R_x(1) & R_x(0) & \cdots & R_x(-p+2) \\
\vdots & \vdots & \ddots & \vdots \\
R_x(p-1) & R_x(p-2) & \cdots & R_x(0)
\end{bmatrix}
\begin{bmatrix}
R_x(k) \\
R_x(k+1) \\
\vdots \\
R_x(k+p-1)
\end{bmatrix}
\]

The impulse response of the \textit{1-step LMMSE predictor} \((k=1)\) filter is then given by:
As it should be clear from its derivation, the implementation of such a filter requires first the wide sense stationarity of the sequence $x(n)$. Moreover, it requires the knowledge of at least $p$ values of the autocorrelation function of the stochastic process $x$ and inverting the $p\times p$ matrix.

To sort out these last issues, an **adaptive** version of LMMSE (the so-called **LMS** predictor) has been proposed and it will be elaborated upon in the next paragraph.

### 3.4.1.2. LMS Predictor

The **LMS** (Least Means Square) algorithm is an adaptive approach which does not require prior knowledge of the autocorrelation structure of the stochastic sequence. Thus, it can be used as an on-line technique for predicting bandwidth. The algorithm scheme is shown in figure 3.12.

![Figure 3.12: LMS algorithm scheme](image)

The filter coefficients $h(n)$ are time varying and are tuned on the basis of the feedback information carried by the error $\varepsilon(n)$. The values of $h(n)$ adapt dynamically in order to decrease the mean square error. Notice that $\varepsilon(n)$, $x(n)$ and $h(n)$ are defined as in the previous section. Indeed, the LMS algorithm is the adaptive version of the optimal LMMSE predictor.

The LMS algorithm operates as follows:

1. initialize the coefficients $h_0$
2. for each new data, update $h(n)$ according to the recursive equation:

   $$h(n+1) = h(n) + \mu \varepsilon(n)x(n)$$

   where $\mu$ is a constant called the **step size**.
If $x(n)$ is stationary, $h(n)$ converges in the mean to the optimal solution $\mathbf{R}_x h = p(k)$ [Hayes 96] [Haykin 91]. In [Haykin 91] it has been shown that LMS converges in the mean if $0 < 1/\mu < 2/\lambda_{\text{max}}$, where $\lambda_{\text{max}}$ is the maximum eigenvalue of $\mathbf{R}_x$. The speed of convergence of the algorithm depends on the value of $\mu$. Large values of $\mu$ result in a faster convergence of the algorithm and in a quicker response to signal changes. However, small values of $\mu$ result in slower convergence and less fluctuations once the convergence is attained. The selection of $\mu$, thus, should be made by trade-off between the above phenomena.

Furthermore notice that, at time $n$, the value $x(n+k)$ is not known, so is $\epsilon(n)$. Thus, the value $\epsilon(n-k)$ is used, instead.

### 3.4.1.3. Normalized LMS (NLMS) Predictor

The normalized LMS algorithm is a modification to the LMS algorithm in which the update equation of the filter’s coefficients is:

$$h(n+1) = h(n) + \frac{\mu \epsilon(n)x(n)}{\|x(n)\|^2} \tag{10}$$

where $\|x(n)\|^2 = x(n)x^T(n)$.

The advantage of using NLMS is that it is less sensitive to the step size $\mu$. According to [Haykin 91], NLMS converges in the mean as long as $0 < \mu < 2$. Again, at time $n$, the value $x(n+k)$, thus $\epsilon(n)$, are not known. In the update equation, the value $\epsilon(n-k)$ is used, instead.

Therefore, the one-step NLMS predictor update equation becomes:

$$h(n+1) = h(n) + \frac{\mu \epsilon(n-1)x(n-1)}{\|x(n-1)\|^2} \tag{11}.$$ 

The computational complexity of a $p$-order NLMS algorithm is that of $2p+1$ multiplications and $p+1$ additions.

### 3.4.1.4. LMK Predictor

The Least Mean Kurtosis Predictor (LMK) is a linear prediction algorithm in which the filter coefficients are determined by minimizing the cost function negated kurtosis [Tanrikulu 94] of the prediction error.

As in previous cases, the problem is to find the impulse response $h = [h(0), h(1), \ldots, h(p-1)]$ of the FIR filter:

$$\hat{x}(n+k) = x(n) \otimes h(n) = \sum_{i=0}^{p-1} h(i)x(n-i)$$
that minimizes the negated kurtosis cost function

\[ J_{\text{LMK}}(h) = 3E^2[\varepsilon(n)^2] - E[\varepsilon^2(n)] \]

\[ = 3E^2[x(n + k) - hx^T] - E^4[x(n + k) - hx^T] \]  

Taking the gradient with respect to the vector \( h \), we obtain:

\[ \nabla J_{\text{LMK}}(h) = 4\{3E[\varepsilon^2(n)]E[\varepsilon(n)] - 4E[\varepsilon^3(n)]\}x \]  

The mean value \( E[\varepsilon^2(n)] \) is estimated through the recursive formula:

\[ G(n) = \beta G(n-1) + (1 - \beta)\varepsilon^2(n) \]  

which gives:

\[ \hat{\nabla} J_{\text{LMK}}(h) = 4\{3G(n) - \varepsilon^2(n)\}x(n)x \]  

Notice that the coefficient vector \( h \) is updated at each time \( n \), hence, for the sake of clarity, its dependence with respect to \( n \) will be explicitly specified as \( h = h(n) \). The parameter \( \beta \) is the so-called forgetting factor and basically represents the pole of the first order IIR filter used to estimate the mean square error of prediction. The update equation of the coefficients vector \( h \) is then (\( \mu \) is, again, the step-size):

\[ h(n + 1) = h(n) - \frac{1}{4} \mu(\hat{\nabla} J_{\text{LMK}}(h(n))) \]  

which can be further normalized as:

\[ h(n + 1) = h(n) + \frac{\mu(3G(n) - \varepsilon^2(n))x(n)}{[x(n)x^T(n)]} \]  

The computational complexity of a \( p \)-order LMK predictor is that of \( 2p+5 \) multiplications and \( p+3 \) additions.

3.4.1.5. Geometric-Predictors

This section deals with prediction algorithms based on the approximation of the behavior of the sequence \( x(n) \) through elementary functions. These techniques do not involve stochastic analysis of time series while they only involve pure geometric considerations.

The basic idea is to determine the \( (p-1) \)-order polynomial interpolator of the set of \( p \) observations \( x = \{x(n), x(n-1), \ldots x(n-p+1)\} \) available. Such a polynomial is of the form:

\[ P_{p-1}(z) = a_{p-1}z^{p-1} + a_{p-2}z^{p-2} + \cdots + a_1z + a_0 \]

and the coefficients \( a_n \) are determined by the condition \( P_{p-1}(k) = x(k), n-p+1 \leq k \leq n \), that is by solving the linear system (with Vandermonde matrix):
Once the coefficients are known, the prediction formula (extrapolation) is as simple as:

\[
\hat{x}(n+1) = P_{p-1}(n+1).
\]

A particular case of polynomial predictor is the so-called **naïve predictor**. This is probably the simplest predictor and is obtained by just considering the order 1 polynomial predictor. The prediction formula becomes as simple as:

\[
\hat{x}(n+k) = x(n).
\]

The predictor is, obviously, unbiased, although it has very low efficiency since its variance is that of the original sequence. In spite of its simplicity, it may be used in many applications where only a rough idea of the future behavior of a stochastic sequence is needed.

### 3.4.2. Chaotic Predictors

The second class of prediction algorithms that will be presented in this document is that of chaotic predictors. The scheme of such algorithms is typically more complex than the ones above described and they are based on the assumption of an underlying chaotic dynamic of the traffic time series. In order to expose the rationale of chaotic prediction techniques, in the following section the basic notions and terminology of chaos are given. Next, the mathematical expression of the Linear Local Predictor and of the Radial Basis Function Predictor will be described.

#### 3.4.2.1. Background on chaotic systems

In this section we briefly introduce the main notions on chaotic systems that will be used throughout the rest of the document. The level of presentation is not intended to be formally rigorous; we rather choose to give the basic ideas in a more descriptive way by avoiding unnecessary mathematical details. For a complete overview on Chaotic systems we refer to [Chua 87].

We begin with the definition of a generic continuous-time autonomous dynamical system, described by the differential equation:

\[
\dot{x} = F(x) \quad x(t_0) = x_0
\]  

(18)
where $x(t)$ is the state of the system at time $t$ and $F : \mathbb{R}^n \to \mathbb{R}^n$ is the field vector. The solution $\phi_t(x_0)$ to (18) is called trajectory and the map $\phi_t : \mathbb{R}^n \to \mathbb{R}^n$ is called flow of the system.

Dynamical systems can be classified on the basis of their steady state behavior, namely their asymptotic behavior as the time $t$ approaches $\infty$.

Although there is no generally accepted definition, chaos can be intuitively defined as a bounded steady-state behavior that cannot be associated with equilibrium points, periodic or quasi-periodic solutions. Chaotic systems have the important property of sensitive dependence on initial conditions (SIC), that is, starting from arbitrary close initial conditions, trajectories diverge significantly until they become almost uncorrelated. According to this property, in chaotic systems the unavoidable effect of the uncertainty of initial condition is magnified and the system behavior becomes unpredictable. In this way, even very simple systems exhibit a quite complex and random like behavior.

### 3.4.2.1.1. Strange Attractors

In spite of their unpredictable evolution, trajectories of chaotic systems are bounded (by definition) and converge in the state space to an object called attractor. The notion of attractor is based on the definition of limit points and limit sets.

A point $y$ is a limit point of $x$ if, for every neighborhood $l(y)$, the trajectory $\Phi_t(x)$ enters $l(y)$ repeatedly as $t \to \infty$.

The set of all limit points of $x$ is called Limit set $L(x)$ of $x$. Limit sets are closed and invariant under $\Phi_t$, that is: $\Phi_t(L(x)) = L(x)$.

A limit set $\alpha$ is attracting if there exists an open neighborhood $U$ of $\alpha$ such that $\forall x \in U, L(x) = L$. The union of all such neighborhoods $B(\alpha)$ is called basin of attraction of the attracting set $\alpha$. In other words, any trajectory starting in $B(\alpha)$ ends up in the attractor $\alpha$.

Typical attractors are points (for stable equilibrium points), cycles (for periodic solutions) and torus (for quasi-periodic solutions). Attractors for chaotic systems are geometric objects more complex than cycles or torus and possess fractal properties. For this reasons they are typically referred to as strange attractors.

Attractors can be characterized by their dimension. The definition of dimension follows the definition of manifold in differential topology provided that the attractor is a manifold. According to it, an attractor has dimension $n$ (or is $n$-dimensional) if, in a neighborhood of any of its points, it resembles an open set of $\mathbb{R}^n$. In this sense, an equilibrium point has
dimension 0, a limit cycle has dimension 1 and a torus has dimension 2. On the contrary, the
gometry of strange attractors do not look like any Euclidean space. Therefore they are not
manifolds and do not have an integer dimension. There are several ways of generalizing the
concept of dimension to a fractional number, which lead to different definition of dimensions
such as: capacity dimension, Information dimension, Correlation dimension and Lyapunov
dimension.

3.4.2.2. The prediction problem
In the above paragraphs, we have been addressing the standard problem in dynamical systems
of, given a non-linear map, determining the asymptotic behavior of iterates. We are now
interested in the inverse problem, that is, given a finite number of iterates, infer the non-linear
map that produces them. This map would then be used in a predictive model. In other words,
given a set of \( N \) observations \( x_n, 1 \leq n \leq N \) and assuming a deterministic (unknown) relation
\( x_{n+1} = F(x_n) \), we look for a map \( f_N : \mathbb{R}^n \to \mathbb{R}^n \) for which:

\[
x_{n+1} = f_N(x_n), \quad 1 \leq n \leq N-1.
\]  
(19)
The problem can be in general solved according to various interpolation techniques; the
resulting \( f_N \) is referred to as chaotic predictor.

3.4.2.3. Metrics for quality of prediction
The large number of possible interpolation techniques that can be used to solve the inverse
problem, require specific metrics to test the effectiveness of predictors. Thus, in order to
 quantify the goodness of a predictor \( f_N \), we introduce the predictor error \( \hat{\sigma}(f_N) \) as:

\[
\lim_{M \to \infty} \frac{1}{M} \frac{1}{V} \sum_{n=N}^{N+M-1} \left\| x_{n+1} - f_N(x_n) \right\|^2
\]  
(20)
with the normalizing factor:

\[
V = \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} \left\| x_m - \left( \lim_{M \to \infty} \frac{1}{M} \sum_{m=1}^{M} x_m \right) \right\|
\]
Values of \( \hat{\sigma}(f_N) \) close to 0 indicate good performance of the predictor. Performance gets
worse as \( \hat{\sigma}(f_N) \) tends to 1. Indeed, \( \hat{\sigma}(f_N) \) close to zero implies predicted values nearly
identical to the actual ones [Casdagli 89]. This condition is excessively restrictive. It is often
enough to require a reasonably low value of \( \hat{\sigma}(f_N) \) and a good match of the statistics of actual
and predicted data, such as auto-covariance function, distribution, variance, and so forth.
3.4.2.4. Prediction algorithms

In this section we describe a method to make predictions about chaotic time series. The first step is to consider the time series as obtained by sampling a continuous time scalar variable \( x(t) \) representing the dynamic of system whose underlying dynamic is that of a strange attractor lying on a \( D \)-dimensional invariant manifold.

Assume then a sequence of observation \( x_n = x(n \tau) \); the sequence is referred to as chaotic time series and the sampling period \( \tau \) is called delay time.

The second step is to embed the sequence in a \( m \)-dimensional state space. The value \( m \) is called embedding dimension. If the strange attractor \( \alpha \) lies on a \( D \)-dimensional invariant manifold, then, necessarily, \( m \geq D \). On the other hand, Takens theorem [Takens 81], assures that \( m \leq 2D+1 \). The state vectors will be of the form:

\[
\begin{pmatrix}
  x(n\tau) \\
  x(n\tau - \tau) \\
  x(n\tau - 2\tau) \\
  \vdots \\
  x(n\tau - (m-1)\tau)
\end{pmatrix}
\]  

(21)

The third step is to select a proper interpolation function \( f_N \) such that:

\[
x_{n+1} = f_N(x_n)
\]  

(22)

Notice that, with our choice of state (21), the future state \( x_{n+1} \) has the first component only unknown.

The prediction procedure involves a number of correlated problems. Indeed, once the prediction function is chosen, and a trial value of \( m \) is selected, the prediction performance is assessed by computing the prediction error. If the results are reasonably accurate (prediction error close to 0), then the prediction problem is solved. Otherwise, a new value of \( m \) is selected and the procedure starts over, until convergence.

3.4.2.4.1. Approximation techniques

So far we have been discussing about interpolation functions, namely maps \( f_N \) such that: \( x_{n+1} = f_N(x_n) \), \( 1 \leq n \leq N-1 \). In many practical contexts, including teletraffic studies, this constraint may be significantly relaxed by allowing \( f_N \) to be an approximant, that is:

\[
x_{n+1} \approx f_N(x_n), \quad 1 \leq n \leq N-1
\]  

(23)

This, obviously, widens the number of techniques that could be used. Typically, they belongs to two different categories: global techniques and local techniques. In the former, the approximant (or interpolant) is applied to the whole series of vectors that can be constructed.
through the sequence. In the latter case, instead, the prediction is made on the basis of the states in the past that are "near" to the current one (neighbors). The rationale of this technique is simple and intuitive: look for pieces of the trajectory in the past that "resemble" the current piece of trajectory and infer the future behavior according to the way the system evolved in that analogous past condition. Incidentally, this method looks well tailored to the case of network traffic which has widely proven to be self-similar.

The selection of the number $k$ of "neighbors" in the past is another critical issue. Typically $k=m+1$ is assumed [Casdagli 89, Farmer 87].

Global techniques are computationally too expensive and are not considered in this presentation. In the next section two local techniques based on linear extrapolation and radial basis functions will be shown.

3.4.2.4.2. Linear Local Predictor

An intuitive approach to express the approximant $f_N$ in (23) is to consider linear extrapolation. Once the set of $k$ neighbors of $x_n$ is selected, compute the slope of the straight line that connects every pair $(x_n, x_{n+i})$ as:

$$a_i = \frac{x_{n+i} - x_n}{\sqrt{\sum_{j=1}^{m-1} (x_{n+i-j} - x_{n-j})^2}}$$

For each computed slope, the possible evolution of the state is evaluated through linear extrapolation:

$$x'_{n+1} = x'_n + a_i \sqrt{\sum_{j=1}^{m-1} (x_{n+i-j} - x_{n-j})^2}, \quad i=1,2,...,k$$

Finally, the 1-step prediction is obtained by averaging all the possible state evolutions, as:

$$\hat{x}_{n+1} = \frac{1}{k} \sum_{i=1}^{k} x'_{n}.$$  

3.4.2.4.3. Radial Basis Functions Predictor

Given the set of $k$ neighbors $x_{nj}, \ j=1,2,...,k$ of $x_n$, a more refined approach to finding the approximant $f_N$ in (23) is to consider a prediction scheme of the form:

$$\hat{x}_{n+1} = \sum_{j=1}^{k} \beta_j \| x_n - x_{nj} \| + \mu$$

Where, in its general expression, the constant $\mu$ is replaced by a polynomial term (usually not included).
The functions $\phi : \mathbb{R}^+ \to \mathbb{R}$, defined as:

$$\phi(r) = \left(r^2 + c^2\right)^\beta$$

are called radial basis functions. A typical value of $\beta$ is $-1/2$.

In other words, the prediction (27) can be interpreted as a weighted sum of terms in which the contribution of each neighbor depends inversely on their distance to the current state (closer states give a bigger contribution).

The values $\lambda_j, j = 1, 2, \ldots, k$ are determined through the knowledge of the past evolution of the state as:

$$x_{n+1} = \sum_{j=1}^{k} \lambda_j \phi \left( \|x_n - x_j\| \right) + \mu$$

(29)

together with:

$$\sum_{j=1}^{k} \lambda_j = 0$$

(30)

Using a matrix notation, the above system of $k+1$ equations is equivalent to the following system:

$$\begin{bmatrix}
\phi_1 & \cdots & \phi_k & 1 \\
\vdots & \ddots & \vdots & 1 \\
\phi_k & \cdots & \phi_{kk} & 1 \\
1 & 1 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
\lambda_1 \\
\vdots \\
\lambda_k \\
\mu
\end{bmatrix}
= \begin{bmatrix}
x_{n+1} \\
\vdots \\
x_{n_k +1} \\
0
\end{bmatrix}$$

(31)

where

$$\phi_j = \phi \left( \|x(n_j) - x(n_i)\| \right).$$

(32)

### 3.5. Service Differentiation for Audio and Video Streams

The Internet with the current best effort service is facing tremendous pressure from new applications. Interactive voice, audio and video generate traffic with characteristics that differ significantly from traffic generated from traditional data applications. The best effort network is designed for very elastic services that can tolerate loss, jitter, delay and varying bandwidth. The new real-time applications are inelastic in nature and cannot respond to varying network conditions in the same way. These new applications, with their inability to adapt and also their potential of consuming a major part of the network bandwidth, are a threat to the stability of the current network. Traffic that is both delay-sensitive and has a variable bit rate (VBR) is the most challenging to handle. Some help from the network in the form of traffic...
classification and prioritization is needed. Not only will service differentiation help the applications, but it will also help maximizing the utilization of the network. IntServ and DiffServ, are suggested network architectures made to cope with these new and demanding applications. In this setting, admission control is most important. Understanding what the applications can do and what the traffic actually looks like are key elements to providing proper admission control. In this section we first look at the QoS Requirements of audio and video and what the sources can do to cope with different network conditions. Following this, we look at traffic characteristics and different traffic models, and then finally we study admission control.

3.5.1. QoS Requirements

Desired quality of audio and video related to users’ satisfaction is subjective. However, there are certain parameters that can be measured and used to map QoS services with the application’s requirement. Bandwidth (throughput), delay, jitter (delay variation) and loss (error rate), are all QoS parameters relevant to audio and video. In the following we look at typical QoS requirements of audio and video.

3.5.1.1. QoS Requirements for Audio

Applications for audio over IP can be one-way streaming or two-way interactive applications. The latter could either be the special case of speech, that is, Voice over IP (VoIP) or general audio signals. Examples of interactive audio applications could be distributed music playing or other videoconference examples.

Delay, Jitter and Loss

Interactive voice has the most stringent delay and jitter requirement. In interactive use, a one-way delay of 150 ms is usually considered as acceptable for VoIP [G.114] [G.107]. With a delay larger than 250-300 ms a normal conversation is not possible. The jitter should be less than 40ms if not to be detected and a jitter above 75 ms is unacceptable [Miras02]. The delay from end to end is in principle irrelevant for one-way streaming of audio. For general audio signals in interactive applications, the requirements will depend heavily on the specific use. One critical example is interactive music playing for which one-way delay would need to be smaller than 40 ms [XW00].

Loss is another important parameter. The loss must be kept within certain limits. However, it is difficult to decide these limits accurately because they depend very much on the size of the packet and whether the loss happens in bursts or packet loss is scattered. For voice, a loss
below 1% is generally acceptable. By using a codec that can support resilience to packet loss, a loss up to 2% will still be acceptable. Tolerance to delay jitter and packet loss for audio largely depends on the compensation techniques that can be used by the application.

**Bandwidth Requirement**

A voice conversation has bandwidth requirements that are low and predictable with its constant bitrate, which varies with the codec used. For audio, the needed bandwidth is higher and must be kept at a sustainable level because tolerance to a degrading quality is extremely low. The bandwidth requirement for audio and speech are shown in table 3.2 below [Eng03] [He02] [P.833].

<table>
<thead>
<tr>
<th>Audio Format</th>
<th>Typical bandwidth requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Compressed MPEG4 coding (aacPlus) of stereo (2 channels) (Considered as good quality by most listeners)</td>
<td>40-48 kpbs</td>
</tr>
<tr>
<td>Compressed MPEG4 coding (AAC) of stereo (2 channels) (Considered as transparent by many listeners, that is, no perceived difference from CD-quality)</td>
<td>128 kbps</td>
</tr>
<tr>
<td>Compressed Dolby Digital for home cinema on the DVI-Video format (5.1 channels)</td>
<td>384 kbps</td>
</tr>
<tr>
<td>Uncompressed CD-quality for stereo (2 channels)</td>
<td>1.4 Mbps</td>
</tr>
<tr>
<td>Uncompressed Multi-channel high-quality audio on the DVD-Audio format</td>
<td>9.6 Mbps</td>
</tr>
<tr>
<td>Speech codecs</td>
<td>8-64 kbps</td>
</tr>
</tbody>
</table>

**Table 3.2: Audio Formats.**

**Perceived Quality**

The perceived quality of the audio signal by the end listener can be established using standardized perception tests for audio signals [BS.1116] or speech [P.800]. Such tests give "Mean Opinion Scores" (MOS) as single numbers that reflect the perceived quality. There are also methods for predicting the results, to determine an objective MOS value. This is done by comparing a sent and a received signal, and let them be analyzed by a perceptual model called "PEAQ" for audio signals, [BS.1387], and "PESQ" for speech signals, [P.862]. It should be noted that these objective methods do not take the delay, or packet loss, into account. The delay is, however, included in another prediction method, called the E-model, which gives a so-called estimated MOS-value [G.107].

**3.5.1.2. QoS Requirements for Video**

Real time video applications are divided into interactive video and video streaming.
Interactive video naturally will have tight latency requirements similar to audio. Streaming video is real-time transport of live or stored video. Delay requirements are here not the main issue but similar to audio, the user will demand a high visual quality where the bandwidth requirements could in the extreme case (thought not very likely) be up to 1.5 Gbps. In order to meet the expectations of users’ experiences with similar non-IP high-quality services, a sustained high bandwidth with very low packet loss must be maintained.

Typical bandwidth requirements for video are shown in the table below [Miras02].

<table>
<thead>
<tr>
<th>Video format</th>
<th>Typical bandwidth requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncompressed HDTV</td>
<td>1.5 Gbits/sec</td>
</tr>
<tr>
<td>HDTV, Interim format</td>
<td>360 Mbit/sec</td>
</tr>
<tr>
<td>Standard Definition TV (SDTV), SMPTE</td>
<td>270 Mbit/sec</td>
</tr>
<tr>
<td>Compressed MPEG-2 4:2:2</td>
<td>25-60 Mbit/sec</td>
</tr>
<tr>
<td>Broadcast quality HDTV (MPEG-2)</td>
<td>19.4 Mbit/sec</td>
</tr>
<tr>
<td>MPEG-2 SDTV</td>
<td>6 Mbits/sec</td>
</tr>
<tr>
<td>MPEG-1</td>
<td>1.5 Mbits/sec</td>
</tr>
<tr>
<td>MPEG-4</td>
<td>5 Kbits/sec - 4 Mbits/sec</td>
</tr>
<tr>
<td>H.323 (H.263)</td>
<td>28 Kbits/sec - 1 Mbits/sec</td>
</tr>
</tbody>
</table>

Table 3.3: Video Formats.

Acceptable data loss depends on the application technology. When the video session is interactive, the user will tolerate some distortion as a trade-off for timely delivery. Compensation techniques used by the applications will improve network utilization and application quality. For streaming video, high values of jitter up to 500ms are considered acceptable [Miras02].

3.5.2. Audio and Video End System

It is apparent that with all these new and demanding applications, if all these different applications are treated in a best-effort network and in the same manner, it will be difficult to provide the needed QoS. The network itself must improve beyond best effort. However, for efficient use of the network, the sources must also take some of the burden. In the following, we look at what the sources can do in order to maximize the perceived quality for a given bandwidth and explore new coding techniques.

3.5.2.1. Audio

Audio coding has traditionally used waveform coding which attempts to recreate the audio waveform as accurately as possible whereas speech coding techniques have used linear
predictive coding that will not recreate the waveform exactly [Eng03]. Recent advances in audio coding have led to methods that use parametric coding, synthetic audio coding and so-called spectral band replication [Eng03].

Layered coding has been explored when suggesting new coding techniques where one approach combines a low-bitrate speech coder for a base layer, and an enhancement layer supplies an extended frequency range and higher quality, that is, a perceptually smaller difference from the original, sent signal [Eng03]. Another approach for layered coding has been suggested which uses a very fine-grain scalability, in steps of 1 kbit/s, based on MPEG-4 coding elements, and in addition offers the possibility to reconstruct the original signal exactly via another (larger) enhancement layer [Eng03]. Some new multi-channel audio formats have hierarchical structures which support layered coding. In that case, a base layer would give a limited spatial quality, that is, a limited sense of direction to the sound sources, whereas enhancement layers would give improved spatial quality [GC04].

### 3.5.2.2. Video and Compression Techniques

Efficient representation and compression of video sources is essential for delivery over a resource-constrained packet-switched network like the Internet. Video coding basically consist of removing two types of redundancy from the original video signal, namely spatial and temporal redundancy. Temporal redundancy, the correlation between adjacent video frames, is removed by motion estimation and compensation. Spatial redundancy, the similarities between neighboring pixels in the texture, is removed by decomposing the signal in the frequency domain using some transform (for instance DCT or wavelet). A short review of image and video coding techniques is given in [EK98]. This hybrid approach of using motion compensation and texture coding in representing a video frame results in different types of frames:

- **Intra frames:** includes texture data only
- **Predicted frames:** includes residual texture data and motion vectors
- **Bi-directionally predicted frames:** includes residual texture data and two sets of motion vectors.

As we can see, the representation of a video frame then consists of *texture information* and *motion vectors* in addition to essential *header* information.

The sizes of the Intra-coded I-frames are usually between 5 and 10 times larger than that of the predicted P-frames, while the size of the bi-directionally coded B-frames are typically half the size of the P-frames for the same PSNR-level compared to the original video sequence.
Compression techniques for video, can either be non-scalable or scalable, where scalable representations are desirable to accommodate the delivery of a single video source to a variety of clients on different access networks.

**Classical non-scalable video coding**

All standardized compression schemes for video up to now have used block-based hybrid motion-compensated DCT-transform coding; ITU H.261, H.263, H.264 and MPEG-1, MPEG-2 and MPEG-4 [RB02]. The baseline versions of these standards (that have gained the most popularity) all result in a non-scalable representation of the video source. This means that all information belonging to a representation of a frame must be provided to the decoder in order to reconstruct the frame without errors. In the case of packet loss, the reconstructed frame will contain errors depending on the type of information that has been lost. Errors in the reconstruction will propagate all the way to the next I-frame because of the prediction-based differential coding. The GOP size that gives the distance between I-frames therefore clearly gives a trade-off between robustness to packet-loss and coding efficiency.

The most widely used formats for streaming video on the Internet is however not standardized formats; Microsoft’s Windows Media format [WinMed] and Real Network’s RealVideo format [RealNet] [Co01]. Both of these are proprietary formats and it is well known that both of the systems produce non-scalable video streams. Real Networks use a simple approach called “stream switching” to facilitate automated choice of best possible video or audio stream to deliver to the client. The video source is encoded at different target bitrates, put into the same file, and the stream that best targets the available bandwidth is chosen for delivery.

**Scalable video coding**

Scalable coding involves producing several different partitions that represent the source or a representation from which several representations can be made through simple processing. This is useful since prioritization schemes and adaptation to meet varying bandwidth conditions can be implemented fairly easily.

**Types of scalability**

Scalability for video can be classified into three separate classes, as discussed in the following. Note that combinations of these are possible.

- **SNR scalability.** Also referred to as quality scalability, deals with the SNR (Signal-to-Noise Ratio) of the decoded video.
- **Temporal scalability.** As the name suggests, deals with being able to adjust the frame rate of the video.
• Spatial scalability. This form of scalability allows for adjustment of the image sizes (resolutions) of the video frames.

Layered video coding

Source coding techniques where the resulting bitstream can be partitioned into a base layer and one or more enhancement layers are called layered coding schemes. Such schemes are supported by the standards MPEG-2 [ISO/IEC00] and MPEG-4 [ISO/IEC01], amongst others. Historically, work on layered coding techniques has been inspired by the pyramidal/ multi-resolution approach introduced by Burt and Adelson [BA83]. For a survey of different approaches for generating layered bitstreams, see [Canne96] chapter 2.

SNR-scalable video coding (Fine-Grain-Scalability)

Most layered coding schemes offer only a very limited number of enhancement layers, making rate adaptation very coarse. Available rates are decided at the time of encoding. This is not the case with FGS, where the enhancement rate can be shaped to meet time-varying bandwidth requirements almost exactly. The FGS extensions of MPEG-4 [ISO/IEC01] support this type of scalability. Details of MPEG-4 FGS can be found in [Li01].

3.5.2.3. Error control for video coding

An error resilient representation is needed to assure graceful degradation in the (inevitable) presence of packet loss and transmission errors. Error resilience tools, or methods for enhancing the error robustness of compressed video for packet-switched transport, can include such different approaches as deciding the optimal coding mode, intelligent packetization, data partitioning, header replication, Multiple Description (MD) coding, combined source-channel coding, forward error correction (FEC), intelligent packet retransmission schemes, congestion control mechanisms and error concealment strategies. In general, all schemes for enhancing the robustness of multimedia communication systems depend on adding redundancy to the encoded bitstream to facilitate a gracefully degrading system performance in the presence of packet loss. The exception is receiver side strategies for intelligent error concealment during decoding. [WZ98a] gives an overview of methods for error control and error concealment in video communication.

Application level tools for error resilience

To prevent packet loss from corrupting the entire video bitstream, start codes are always used in the beginning of a video frame and slice (a line of 16x16 image blocks in a video frame) to regain synchronization in the event of information loss. In general, resynchronization markers
can be used extensively in the compressed bitstream, the trade-off is higher overhead versus better error resilience.

Other tools for increasing the error resilience of compressed bitstreams are robust entropy coding by using reversible VLC codes (RVLC) and header extension codes that replicates header information in the bitstream. The latter approach is motivated by the fact that loss of header information will have a devastating effect on reconstructed video quality. Video data partitioning and robust packetization of compressed video data are approaches that do not necessarily introduce a lot of overhead, and are important and effective approaches in increasing the video application's resilience to packet loss. [WCX01] [RB02] [WZ98a] are just a couple of references considering error resilience in networked video applications, and the tools mentioned above have also been included in the MPEG-4 video standard and the ITU H.263 standard [RB02].

**Forward Error Correction**

Error correction codes are, as the name implies, designed to detect and correct errors in case of transmission errors. This obviously comes at a price of added overhead/redundancy. The FEC coder produces $n$ packets from $k$ source packets. The receiver can then reconstruct the source data exactly by receiving any subset of $k$ packets (or more). The perhaps most used error correction code to this end is the Reed-Solomon (RS) code. A good review of packet loss recovery techniques for audio and video can be found in [PHH98] and [WZ98b], respectively. [TZ01] discusses the use of scalable coding and FEC in layered multicast for obtaining different protection levels for different receivers. In [Puri01], FEC is used for converting a single description code into a Multiple Description code for error recovery and resilience.

**Multiple Description (MD) coding**

Multiple Description coding produces a non-hierarchical representation of the source. This implies that every description (i.e. packet) is useful in itself, and not dependant on others. By adding redundancy amongst descriptions, estimates of the lost data can be generated. An excellent overview of MD techniques and their applications can be found in [Goyal01].

**Error Concealment**

Error concealment strategies can be employed in the receiver end to often quite effectively mask the loss of information due to packet loss. In general, the problem is how to recreate those parts of the visual representation that are lost or distorted because of packet loss and information loss. The algorithms are typically implemented in the video decoder, and can be
divided in two categories based on which domain is being used for masking; *temporal* and *spatial* interpolation. Temporal error concealment methods typically employ information from adjacent frames that has been reconstructed without any error to estimate lost information in the present frame. Spatial methods use information from the same frame to spatially interpolate lost information [WZ98a].

### 3.5.2.4. DiffServ Network and Video Coding

A DiffServ network can give different types of traffic different services depending on the QoS requirements. In this way, bandwidth demanding sources like high quality video cannot starve bandwidth for an urgent interactive conversation. However, giving applications that require high bandwidth its own service class will not utilize the total bandwidth very efficiently. The simplest way of doing service differentiation of a video stream, is just to give the different I, P and B MPEG coded frames different priorities and thereby enhancing the use of network resources [ZAS02]. By using layered coding techniques and give the different layers different priority depending on their importance will give more flexibility.

In the literature, there have been several approaches considering the delivery of video over differentiated service networks. A recent one by Shin, Kim and Kuo [SKK01] presents a framework for utilizing the different service levels of DiffServ in delivering packet video. Different parts of the compressed video are categorized in different classes based on a priority index, and the video packets in the different categories are mapped to service levels in DiffServ. The categorization scheme prioritizes groups of macro-blocks (MB) that are Intra-coded, and MB’s in the predicted P-frames that have large motion-vectors associated with them. The latter point is appreciated when considering the potential impact the loss of these motion vectors has on the reconstructed video frame. The higher the motion, the bigger the reconstruction error in case it is lost. When employing error concealment algorithms in the decoder, it is also easier to obtain a good temporal interpolation in low-motion areas than in high-motion areas. [Zhang02] is considering the same aspects in streaming of MPEG-4 video over DiffServ enabled networks.

In [SZZ01], Shao, Zhu and Zhang present a user-aware object-based video transmission system that incorporates a new transport framework, service differentiation functionality both within an IP session and amongst different IP sessions in transporting the different video objects. In addition to the possibility of giving more important video objects a higher priority, [SZZ01] also considers bitstream classification for individual video objects, and the
corresponding prioritization and packetization of classified video data. The proposed solution also includes object-based rate control by frame dropping in a DiffServ scenario.

For multicast application scenarios, [LiLiu03] provides a survey of multirate video multicast solution applicable to differentiated service networks, while Li, Liu and Y-Q. Zhang surveys solutions for adaptive video multicast in [LLZ03]. [LiLiu03] classifies multirate multicast applications into three categories, namely the stream switching approach they call “stream replication”, layered multicast and agent-based multicast. The layered approaches can further be separated in cumulative and non-cumulative approaches, depending on whether the different bitstream layers are hierarchically dependent of each other or not, that is, is the complete base layer needed to properly decode enhancement layers or not. Within this classification, [LiLiu03] discusses factors like bandwidth economy, adaptation granularity and the coding complexity and efficiency.

3.5.3. Traffic Characterization

The network alone cannot provide QoS guarantees to multimedia applications such as Internet audio and video. The behavior of a traffic source can affect the quality of service delivered to it significantly. Conversely, such applications should not be admitted if their QoS requirements cannot be met, or if they destroy guarantees already given. Clearly, the characterization of a traffic source plays an important role for service guarantee and admission control.

Source traffic characterization aims at characterizing the source in a general descriptive way [HS92] [Adas97] [IR99]. A video source produces cyclically the frames in a Group of Pictures, and the distribution of the various frames (I, P, and B) is typically described together with the use of second order statistics to describe dependencies. For an audio source the encoder produces equidistant samples which can be described likewise. When layered coding is employed, the various layers need to be described in a similar way.

The frames or samples need to be packetized before they are sent over the IP-network, and this process needs to be known to deduce a characterization of the packet stream, or alternatively, the packet stream itself can be studied. Needless to say, the packetization and the assignment of priorities greatly impacts how the packets are generated at the source.

Much work was done in the 1990s to characterize MPEG-1/2 traffic. Not much is found on MPEG-4. The work by Fitzek and Reisslein [FR01] reports on a study of frame size traces generated by MPEG-4 and H.263. It contains some important references.

Very little is found on characterization of high quality audio traffic, most studies are on VoIP.
3.5.3.1. Traffic Models

Traffic models of audio and video sources are used for simulation or for analytical studies. Models for analytical studies proposed in the literature are stochastic models of various types, e.g. Markov-modulated rate or Markov-modulated Poisson processes, and Gaussian traffic model. There is a rich literature on such modeling techniques and associated solution techniques. Parsimonious models are needed for getting tractable solutions, but even then parameter setting is not trivial. (And as pointed out earlier traffic characterization for new applications is lacking.) Exact solutions techniques usually exist for sources feeding just one node. Good models for aggregated traffic are lacking (to be discussed later), making it hard to get analytical results for end-to-end properties of traffic through a network. The end-to-end properties obviously depend on which type and how much traffic (i.e. flows or sessions) are admitted. Admission control is therefore pertinent. Such control can either be based on models or on live measurements. Both audio and video traffic can live with soft (stochastic) guarantees.

By use of some traffic constraint function, a bounding traffic model may be defined. In such a model, the traffic constraint function is determined by a (usually small) set of parameters or functions, termed as traffic descriptor, to bound traffic. This makes it easier to give service guarantees for a flow through a network, but may result in a poorer utilization of the links.

Stochastic Traffic Models

Aiming to capture as accurately as possible the statistical characteristics of real audio/video traffic, stochastic traffic models have attracted much research attention. Many such models have been proposed in the literature and several reviews are available [HS92] [Adas97] [ML97] [AZN98] [IR99]. Some examples of stochastic traffic models are Markov-modulated Poisson processes, Regression models, Long-Range dependent (LRD) models and Gaussian models [AZN98]. The ON-OFF model is perhaps the most popular source model for voice. Regression models have been widely used for modeling video traffic [Adas97], [IR99]. Markov-modulated and regression traffic models are short-range dependent traffic models. However, audio and video traffic usually have a variable bit rate (VBR) and long range dependence is intrinsic to VBR sources [Beran95]. Due to the central-limit theorem, the Gaussian model has been used to characterize the behavior of traffic aggregate [AZN98].

While stochastic traffic models can characterize the traffic accurately, they do not easily lend themselves to on-line resource management such as admission control. This has led to
research on bounding traffic models [Zhang95], for resource allocation and admission control of audio and video applications.

**Bounding Traffic Models**

For each bounding traffic model, the exact traffic pattern for a flow is unknown. The only requirement is that the amount of the traffic is bounded in certain ways. An interesting and important property of such a model is that the superposition of flows represented using this model can still be characterized using it and the output traffic of a node can be represented using the same model. In general, there are two types of bounding traffic models, the deterministic bounding traffic models and the stochastic traffic models.

One widely used deterministic bounding traffic model is token bucket [Cruz91a]. The token bucket model \((r, b)\) uses two parameters, token rate \(r\) and token bucket size \(b\), to characterize a traffic source. Under this traffic model, the amount of traffic generated by the source in any time period \(t\) is bounded by the function \(r*t+b\). A variation of the token bucket model is the dual token bucket which includes two more parameters, peak rate \(p\) and maximum packet size \(m\). That is, the dual token bucket model has four parameters: token rate \(r\), bucket size \(b\), peak rate \(p\) and maximum packet size \(m\). Under dual token bucket model, the amount of traffic generated by a source during any time interval \(t\) is bounded by \(\min\{r*t+b, p*t+m\}\). Both token bucket and dual token bucket traffic models have been used in IntServ [SW97] and DiffServ [DiffServ] to describe the behavior of a traffic flow requiring deterministic service guarantees. Another further extension of the token bucket model, proposed in [Wrege96], is to use multiple \((r_i, b_i)\) pairs to characterize the traffic source, under which the amount of traffic generated in any period \(t\) is bounded by \(\min_i\{r_i*t+b_i\}\).

Besides token bucket, there are several other deterministic bounding traffic models, which include \((x_{\text{min}}, x_{\text{max}}, I, m)\) model [FV90] and Deterministic Bounding Interval Dependent (D-BIND) model [KZ97]. While using different parameters, each of them defines a traffic constraint function that bounds the amount of traffic generated by the traffic source.

A more general deterministic bounding traffic model is the arrival curve [Cruz91a, LBT03]. With this model, a general function \(\alpha(t)\) is used to describe the constraint on the amount of traffic generated by the traffic source. Specifically, it states that during any time period \(t\), the amount of traffic generated is bounded by \(\alpha(t)\). All above reviewed deterministic bounding traffic models can be viewed as special cases of service curves.

There are several stochastic bounding traffic models proposed in the literature which can be used for service differentiation and admission control for audio and video traffic.
The Exponentially Bounded Burstiness (EBB) model [YS93] uses three parameters to describe a traffic source, the upper rate \( r \), decay rate \( \gamma \), and asymptotic constant \( C \). EBB specifies that the probability that the amount of traffic generated by an EBB flow is greater than \( r^t+x \) is bounded by an exponential function of \( x \) with asymptotic constant \( C \) and decay rate \( \gamma \).

The Stochastic Bounding Interval Dependent (S-BIND) [ZK94] model extends D-BIND, using a family of two-tuples \(((R_{t1}, t1), (R_{t2}, t2), \ldots)\) to characterize a traffic source. With this model, the amount of traffic generated by the source over any interval of length \( t_i \), is stochastically smaller than \( R_{t_i} \).

The Stochastically Bounded Burstiness (SBB) model [SS00] extends EBB. SBB uses an upper rate parameter \( r \) and a bounding function \( f \) to describe a traffic source. While under EBB this bounding function has an exponential form, it can be any one within a general function family under SBB. As a special case of SBB, Weibull Bounded Burstiness (WBB) model [Yu01] is proposed to study long range dependent traffic. Under WBB, in addition to the three parameters of EBB, it has an index parameter that can be further expressed in terms of the Hurst parameter commonly used to characterize the degree of long range dependence of traffic.

While all the above bounding stochastic traffic models characterize a traffic source based on the amount of traffic generated over a time interval, there is another class of stochastic bounding traffic models that do so based on the virtual queue process of the source. Particularly, effective bandwidth (also called equivalent capacity) model [PE96] [WS99] and generalized SBB (gSBB) [Yin02] belong to this class.

Like SBB, gSBB also uses two parameters, an upper rate \( r \) and a bounding function \( f \), to describe a traffic source. While it is the amount of traffic that is stochastically bounded by the bounding function in SBB, in gSBB, it is the buffer size of a virtual single server queue system with constant service rate \( r \) that is stochastically bounded by the bounding function. The effective bandwidth model is a special case of gSBB, which has an exponential form for the bounding function. A similar model to gSBB, called Probabilistic Burstiness Curve (PBC) has been used in [CL97] to characterize video sources.

**3.5.3.2. Open Challenges**

As reviewed above, sophisticated models for audio and video traffic sources abound, and their behavior through a single (network) node is relatively well understood, as it has been (and
continues to be) the subject of extensive research for the last several decades [HS92] [Adas97] [AZN98] [ML97] [AZN98].

Little, however, is understood about how the traffic from such a single (audio or video) source is shaped due to multiplexing and switching through a series of nodes in the network. Clearly, a good understanding of the impact of traffic shaping through the network nodes is key to the design of robust and efficient admission control procedures. (Resource reservation at an internal network node on behalf of a newly arrived connection requires the characterization of that traffic source as well as the characterization of the other aggregate traffic at the input of that node!)

To this end, it is important to develop good and realistic models for the aggregate of heterogeneous traffic sources and to understand the behavior of such aggregate traffic as it traverses (multiplexed and de-multiplexed) one or a series of network nodes. The properties of the aggregated traffic depend on the traffic mix and the service differentiation in the routers. Audio and video traffic compete individually, as they also do with other traffic like TCP-traffic. Most current differentiation schemes make it difficult to assess the end-to-end properties in an analytical way. To be able to do that, admission control and service differentiation -and even possibly traffic separation-, must be done a smart way. Work by Kortebi [KQR] suggests solution in that direction.

As pointed out earlier, new audio and video solutions will generate new types of traffic; measurement data for these are missing, and new source models/new parameterization is needed. This, together with new ways of handling the traffic, requires new models for the aggregated traffic.

Although rigorous or approximate analysis of the above is sought, it may be very difficult, if at all possible. A feasible and promising alternative, however, is to gain insight and understanding of the above through extensive measurements, realistic traces and simulation studies and laboratory setups.

Evidence and possible causes of long range dependence (LRD) and self-similarity in single (and aggregate) video and Internet traffic sources have been reported and discussed extensively in recent literature [Leland94] [GW94] [Beran95] [PF95] [ENW96] [CB96] [Norros94]. Heavy-tail distributions characterizing file transfers have also been observed and the analysis of some simple models has been considered [Cohen98] [BD98] [BC99] [BBJ03]. While the impact of these sources on network traffic (and, hence, performance) is undisputable, there is still much to learn and understand about their aggregate behavior, their
shaping through network nodes, their impact on each other and on other (conventional) traffic sources, etc.

Only a few traffic (performance) models involving self-similar arrival processes and/or heavy-tail service time distributions can be found in recent literature [Norros94] [Cohen98] [BD98] [BC99] [BBJ03]. Much more must be done to gain a better insight and understanding of these traffic sources. Needless to say, many of the questions raised above (for conventional traffic sources) are also, if not more, valid in the presence of self-similar traffic and heavy-tailed distributions. How important it is to model this LRD for traffic where short buffers are needed may be questionable.

3.5.4. Admission Control

Networked audio and video applications usually have stringent QoS requirements. In order to ensure that the QoS provided is guaranteed by the network, admission control is applied in addition to proper resource reservation or allocation at the corresponding network elements. Admission control ensures that a newly admitted flow can receive its required service guarantees. In addition to this, it also makes sure that the already admitted flows will not receive services below their required quality guarantees. In the literature, there have been a large number of results for admission control (AC), which can be classified into different categories using different criteria.

First, based on the type of required service guarantees, an admission control algorithm is either for deterministic service guarantees (hard guarantees) or for stochastic service guarantees (soft guarantees). Usually, for deterministic service guarantees such as Guaranteed service in IntServ and Expedited Forwarding service in DiffServ, deterministic bounds (on delay, jitter or loss) are calculated based on some specific traffic characterization such as token bucket constrained. Admission control tests are further applied based on such calculated bounds, e.g., [CL95]. Since deterministic service guarantees have given derived bounds, admission control is relatively easy and not so many explicit admission control papers fall into this category.

Most admission control schemes in the literature focus on stochastic or statistical service guarantees. Audio and video applications, depending on the use of compensation techniques, can tolerate some small level of service degradation. As a result, stochastic or statistical service guarantees can provide acceptable quality of service to most audio and video applications and at the same time improve network efficiency tremendously compared to
deterministic service guarantees. Several reviews of admission control schemes for stochastic service guarantees are available, such as [PE96], [SYT99] and [KS99].

Second, an admission control scheme belongs to either model-based (also called parameter-based) or measurement-based groups. In a model-based or parameter-based admission control scheme, an individual flow or an aggregate flow is modeled by some given parameters or functions as introduced in section 3. The scheme makes admission decisions based on the service guarantees derived from these parameters and those required by the flow. Examples of admission control schemes in this group can be found from [PE96] and [KS99].

Recently, measurement-based admission control has attracted a lot of interest, because it can achieve higher network utilization than parameter-based admission control [Jamin97]. Schemes for measurement-based admission control can be divided into two sub-groups. In one sub-group, the measured objects are remaining network resources such as bandwidth; in the other sub-group, the measured objects are parameters for some given traffic models. For example, the scheme proposed in [Jamin97] and most schemes reviewed in [JSD97] and [BJS00] belong to the first sub-group, while the scheme proposed in [QK01] belong to the second group where parameters for traffic envelope, a traffic model similar to D-BIND [KZ97], are measured. To provide good measurements of parameters of interest, different techniques have been proposed, which include point samples [TG97][GT99][GT03], exponential averaging [Floyd96] [Jamin97], frequency-domain approach [LL97], fuzzy logic [Bensaou97], and wavelet approach [RVA00].

Third, based on their supported performance metrics, admission control schemes can be classified into the following groups: 1) rate-based; 2) loss-based; 3) delay-based; 4) mixed. The performance metric for rate-based admission control is the rate or bandwidth. In other words, a rate-based admission control scheme ensures that a certain amount of bandwidth is available to any admitted flow. Schemes proposed in [FV90] and [LLD96] belong to this group.

Loss has been widely used as the performance criterion for admission control. One important development for this is effective bandwidth (also called equivalent capacity) and its extension: large derivation theory [GAN91] [EM93] [PE96] [KS99] [SYT99]. Various other techniques as reviewed above for measurement-based admission control are mainly for loss-based admission control.

While deterministic delay guarantee has been widely used as an important criterion for admission control particularly for Guaranteed service in IntServ and Expedited Forwarding in DiffServ (e.g. [CL95] and [Jamin97]), very few results are available which use stochastic
delay guarantee as the admission control criterion. One recent result for admission control based on stochastic delay guarantee is [GCL04]. Similarly, while there are various admission control schemes that have considered both bandwidth and deterministic delay (e.g. [Jamin97]), or bandwidth and loss (all effective bandwidth based schemes), not much work has been found for admission control which considers both stochastic delay and loss or stochastic delay and bandwidth or all the three as the admission criteria.

Finally, admission control is performed either in a centralized manner or in a distributed manner. In centralized admission control, there is a central admission control manager, which keeps track of the resource usage in the whole network or domain and makes admission decisions. An example of centralized admission control is Bandwidth Broker for DiffServ networks [Teitelbaum99][ZDH01]. While centralized admission control is ideal for possibly optimal resource utilization, it is neither robust nor scalable: any failure or congestion in the central manager or on the link to the manager could make it inaccessible to applications.

Because of the limitations with centralized admission control, distributed admission control has been an active research area in the past few years. In fact, in IntServ networks, admission control is normally performed in the hop-by-hop manner, which is a special case of distributed admission control. In addition, [KKZ00] [Breslau00] [Bianchi01] [CKK01] [BN03] [Bosco03] presented several typical distributed admission control schemes. In [KKZ00] [Breslau00] [Bianchi01], admission control is performed by end users or hosts. In [CKK01], it is done at egress routers. In [BN03], admission decision is made at edge routers and is for core-stateless networks. In [Bosco03], while admission control is also performed at edge routers, it is for MPLS networks.

### 3.5.4.1. Open Challenges

As noted above, accurate characterization of individual sources and aggregate traffic as they traverse network nodes is crucial for efficient resource management and robust admission control. The traffic model of a given source (or aggregate traffic) at the ingress of the network is no longer valid characterization of the same traffic at internal network nodes. Because of this, model-based admission control algorithms may not be expected to perform favorably in real network environments.

Carefully designed measurement-based admission control algorithms, however, have a better chance of achieving `good' performance [SYT99]. This is because resource allocation and admission decisions are based on active on-line measurements of the aggregate traffic at network nodes along the path of the requested connection [GT99].
Still, however, an incoming request for admission must provide a traffic characterization of the associated source, as well as minimum quality of service requirement. While the provided characterization of the source is no longer valid at internal network nodes, the impact of this misrepresentation may be insignificant at nodes switching large volumes of aggregate traffic. But this is not the case at (or close to) the ingress of the network. Novel and creative (model- or measurement-based) methodologies need to be developed in order to enhance the performance of admission control in this regard. Given the lack of results and limited experience regarding the dynamic behavior of aggregate traffic and self-similar traffic, admission control based on measurements and the time-scale decomposition approach [GT03] may hold much promise and should be further explored. Another approach that deserves more attention is that based on aggregate traffic envelopes [QK01], as it caters for a wide range of traffic types. Scalability and dependability issues favor distributed algorithms for network management and admission control [KKZ00]. Distributed, simple, and robust mechanisms are becoming increasingly essential for management and operation of future and large networks - a rich and less explored area of research.

3.6. Pricing Mechanisms

The evolution of the telecommunication and networking fields is happening at a fast pace and concerns both the technologies and the regulation and business aspects. Until recently, the provision of telecommunications services involved only two sides—the user and a single provider. Nowadays, the user is in presence of many choices because of the liberalisation of the telecommunications markets in most countries: there may be several providers offering the same services. Besides, a simple connection can involve more than two operators between technologies and services. From this multiplicity it turns out necessary to find out processes and regulations to price the services with equity. This means to ensure profits to the operators involved with respect to satisfying at maximum the user’s service utility.

The fact that the users have currently many options makes it necessary to take into consideration both the users’ and the operators’ needs simultaneously, and to try to satisfy them both. The optimisation of the telecommunications services should be undertaken as a two-stage problem. The first stage is optimizing the prices properly and the second deals with the reaction of the users with respect to the prices proposed by the operators.
In the last ten years, a great amount of research has been undertaken on pricing. In this section, we summarize a selection of pricing schemes for telecommunication networks, following the outline of [Falkner 00]; for an exhaustive overview on recent methods, the reader is referred to e.g. [Courcoubetis 03] or [Tuffin 03]. In the earliest methods of telecom pricing (like flat-rate pricing), prices are set independently of the consumer’s utility or willingness to pay. More recent methods take into account the user by considering her willingness to pay or some QoS requirements, through either a non-linear utility function (proportional fairness pricing), a system of price-equations (priority pricing) or a pre-defined maximum budget (“smart market”). The bi-level model sets prices that maximize revenue for the operator taking into account user demand.

3.6.1. Flat-rate pricing

Flat-rate is probably the most common pricing scheme in IP networks. Under the simplest flat pricing [Fishburn 98], or pay-in-advance subscription [Anania 97][Falkner 00], the user pays a fixed amount per time unit (say, a month) irrespective of her network usage. In practice, contracts between the user and the network provider are sometimes not on a “all-you-can-eat” basis, but instead introduce an upper bound on the user’s aggregated or maximal network usage; this bound may correspond to a limit on, say, the monthly volume of data uploaded by the user. Users may be subject to “penalties” (which may take the form of a usage-based charge, for example) when the contractual upper-bound is breached.

Remark that flat-rate pricing is not well suited to a network service offering proper QoS guarantees, nor can be used as a traffic engineering tool to deal with congestion. Besides, it is not incentive compatible (it stimulates what has been called browsing mentality, meaning that network usage is independent of the value users put on their traffic) and is unfair, in the sense that light users subsidize heavy users. On the other hand, this scheme is simple and convenient for providers: in particular, from a technical point of view, it is very easy to implement and does not require sophisticated accounting and billing mechanisms.

3.6.2. Paris Metro pricing (PMP)

In the PMP scheme introduced by Odlyzko [Odlyzko 97][Odlyzko 99], the network is split into a few (say, three or four) independent subnetworks or tiers; more precisely, a fixed fraction of the capacity of each link is totally allocated to each subnetwork. Each subnetwork operates in a best-effort way, and traffic is charged on a usage-based basis: the tariff will be different for each subnetwork. Users make a choice between the subnetworks with respect to an expected service level and their willingness to pay, so that (hopefully) congestion will be
alleviated in the most expensive ones, resulting in a better service quality. Note that this method does not offer any QoS guarantees, so it is somehow weak with respect to more complex QoS techniques; however, it is very attractive because of its ease of implementation and management.

Concerning implementation issues, note that PMP can be easily integrated in a DiffServ-enabled IP network. Users may select the desired service class by means of marking their packets with an appropriate DiffServ codepoint (DSCP) defined by the network operator. Then, the operator ensures the separation of traffic in independent classes using the standard classification and scheduling algorithms found in routers.

Despite its appealing simplicity, PMP may suffer from several drawbacks. As pointed out in [Gibbens 00], network segmentation as performed by PMP may not work under competition. Other studies [Ros 04][Hayel 03] suggest also that, in terms of revenues, an operator may be better off by not splitting its network in several best-effort subnetworks.

3.6.3. **Edge pricing**

Edge pricing [Shenker 96] is based on simple expected values of congestion costs. The congestion is a direct consequence of bandwidth scarcity. For instance, if we know how to evaluate the network load at a certain period of a day, then we can derive the respective congestion costs. Thus, an example of an expected congestion cost would be time-of-day charges [Falkner 00]. An estimation of the delay from a source to its destination allows to apply charges, either at the source or the destination. Proper QoS guarantees are possible but the system would then require the use of a CAC (Connection Admission Control) policy.

3.6.4. **Effective bandwidth pricing**

The effective bandwidth pricing scheme [Kelly 94][Kelly 97] allocates at a request origin the quantity of bandwidth needed in order to satisfy the desired QoS. It induces a centralized vision of the network that allows at each instant to evaluate the bandwidth availability. Such a scheme may allow an online treatment of requests. Under effective bandwidth pricing, users are charged for their requests according to a linear function tangent to the bandwidth evolution curve evaluated at the origin of each request. This centralized vision and the fact that it does not take into account users’ willingness to pay deviate this method from the reality. However, proper QoS guarantees can be given. A scheme such as CAC ensures that access to the network is based on resource availability rather than users’ willingness to pay [Kelly 94b]. This pricing scheme can also be extended to allow for time-varying prices.
[Falkner 00]. One disadvantage of the scheme is that the functional form of the effective bandwidth is assumed to be known in advance.

3.6.5. Expected capacity pricing
Under this scheme [Clark 97], users specify the bandwidth that match their QoS requirements. Through contract agreements between users and the network provider, the billing is done with respect to the expected bandwidth that the network can provide independently of usage. Such a system allows to control the traffic and the bandwidth availability, additionally, proper QoS guarantees can be provided.

3.6.6. Priority pricing
Under priority pricing [Cocchi 93][Cocchi 91][Gupta 97][Gupta][Mendelson 90] the traffic in a multi-service-class network is managed in a complete decentralized setting with priority service classes. Pricing is done accordingly to priority classes. This scheme allows, on the one hand, users to evaluate and determine the time, the characteristics and the priority of their services. On the other hand, it permits the provider to control the traffic as a whole. The model then seeks to maximize the collective benefits of the system [Gupta 97][Mendelson 90]. During periods of congestion the network assigns the traffic by priority level. However, proper QoS guarantees are here also not guaranteed.

3.6.7. Smart-market pricing and auctions
The “smart-market” pricing scheme [Mackie 95] is based on an auction scheme when the network is congested. Besides a fixed charge, a usage charge is introduced when the network is congested. Each packet holds the user’s willingness to pay or bid for its transmission. The ”auctioneer” sorts all bids and determines a threshold for the auctions. Such a threshold is a function of the available bandwidth and determines the congestion marginal cost. All requests respective to bids greater than the threshold will be transmitted paying the fixed threshold. Note that accepted packets are not charged their bid, but the so-called second price, that is, the bid of the first rejected packet. It is known from standard economy theory that second-price auctions provide incentives to users to declare their actual valuation. "Smart-market” pricing encourages network efficiency and allows for congestion control, but questions remain open as regards to the ”auctioneer” and the definition of the model itself. The scheme also does not insure service guarantees or even a guarantee of transmission.
Since the pricing scheme in [Mackie 95] is based on per-packet auctions, this method is clearly extremely difficult to implement. A more realistic mechanism has been proposed, called Progressive Second Price (PSP) auctions [Semret 99][Tuffin 02], in which auctions for bandwidth are carried out in a periodic manner. A bid is then composed of the requested amount of bandwidth and the unit price the user is willing to pay. Auctions happen on a flow/user scale and not on a packet scale, so in principle such a mechanism is feasible from a practical point of view; an RSVP-like protocol [RFC 2205] may be used for carrying the signalling associated with bids. Note that a major drawback of PSP has been highlighted in [Maille 03], namely, that the first user entering an auction has the incentive of asking for all the bandwidth at a very high price, so that the next users will be excluded (and the first will get a maximum of bandwidth at a low price). Maillé and Tuffin [Maille 03b] propose the introduction of a “sanction bid” to alleviate this problem. Another extension [Maille 04] of the original PSP system consists in introducing the notion of multi-bid auctions. In the multi-bid scheme, each user submits a set of (bandwidth, price) pairs, from which the network can compute a pseudo-demand function and hence a pseudo-market clearing price. The amount of bandwidth allocated to each accepted user will be close to her demand at the market-clearing price. The main advantage of this scheme, with respect to the classical PSP, is that it is a one-shot mechanism—the signalling overhead is much lower than in PSP [Maille 04].

3.6.8. Responsive pricing

This pricing mechanism also only comes into operation during periods of congestion [Murphy 95b] [Murphy 97]. It works in a dynamic setting [Murphy 95]. Prices are modelled as an increasing function of the bandwidth scarcity. In [Falkner 00], this pricing scheme is classified as being socially fair and compatible with existing technologies.

3.6.9. Proportional Fairness Pricing

Proportional Fairness Pricing [Gibbens 99][Kelly 97b][Kelly 98] intends to fairly allocate network resources as regard to the users’ willingness to pay. The model defines network link prices and ties them to be Lagrangian multipliers. In particular, a proportionally fair resource allocation $h^*$ is one that satisfies for all users, $r \in R$, with a willingness to pay $w_r$, the inequality

$$\sum_{r \in R} w_r \frac{h^*_r - h_r}{h^*_r} \geq 0$$
for all feasible allocations \( h_r \). This inequality amounts to the optimality condition of the following optimization program:

\[
\max \sum_{r \in R} w_r \log h_r
\]

such that \( h_r \) is feasible with respect to user demand, the capacity constraints on the network links are satisfied, and flows are non-negative. Thus, the Lagrangian multipliers on the link capacity constraints are then interpreted as the added costs that would drive the allocation to proportional fairness, without imposing them explicitly.

Under this scheme the users’ utilities and network profit are maximized. Note however that \textit{prices are endogenous and cannot be varied to maximize network profit}. Congestion in this scheme is avoided by allocating resources to the users, and billing and measurements are not required [Falkner 00]. There is however no priority mechanism, and during periods of high demand each user would get a proportionally smaller amount of bandwidth. In the former case nothing guarantees the satisfaction of the users.

Proportional fairness pricing is of significant interest for the allocation of telecommunication resources, as it provides an economic foundation to the resulting allocation, when a utility function is used to describe user preferences in the place of the traditional (linear) cost minimization objective. It further provides a starting point for modelling link pricing and its relation to the demand for resource usage on a telecommunications network. That is, it combines marketing ideas of pricing with technical preoccupations of flow levels on the network. However, this model of Proportionally Fair prices and resource allocation does not go as far as permitting the network manager to treat pricing as a goal in itself since the decision variables are all resource allocation levels, and prices are outputs of the algorithmic strategy.

3.6.10. A restricted bi-level pricing

The restricted bi-level pricing scheme proposed in [Bouhtou 03][Diallo 03] is closely related to Proportional Fairness pricing. It extends and generalizes it to a model that explicitly considers the network manager’s objective of maximizing revenue.

The model provides a single saddle-point problem, which is an important special case of the general bi-level programming problem. It permits optimizing the operator’s objective function over a restricted set of prices. Such a set is derived from the Karush-Kuhn-Tucker optimality conditions of a resource allocation problem (a multi-commodity flow problem over the network, subject to capacity constraints). The model further draws a parallel between the method of marginal-cost pricing, and operator’s profit maximization. In addition, this
approach has the advantage of providing a model having a unique optimal solution, and is therefore computationally very easy to solve, as opposed to its more general bi-level counterpart.

3.6.11. Pricing for guaranteed performance services

Today's Internet resources are equally distributed making the only one possible service be a best effort service. With the growth of Internet, several traffic flows belonging to new sophisticated applications have some specific requirements and need some guarantees of QoS. These services can pay to get a level of QoS when best effort still guarantees connectivity to classical traffic. The question that arises is how to price the different proposed QoS levels in an efficient way.

Overview

Some pricing models have been proposed in the context of QoS-enabled architectures including ATM, MPLS and QoS over IP (Integrated services, RSVP). In such models, resources are negotiated and priced at a connection setup time-scale in order to meet QoS requirements. Those pricing models can be classified according to the overlay architecture: ATM [LV93, AL97, MM94b, MM94a], MPLS [BR03], Intserv [PKF92, FSVP98]. However, some mechanisms are not specific to a particular architecture and can be applied to any environment where resources are allocated in advance to guarantee QoS requirements. Furthermore, prices can be static and only influence admission control. In other cases, prices are dynamic for instance in [PT00] they correspond to the congestion state and users react with new demands. Here, we rather propose a classification of some of those pricing models with respect to the main involved concepts: adjustment schemes, effective bandwidth and auctions.

Adjustment schemes

Pricing Models that use adjustment schemes come with different flavors [CSS96, LV93, SFY95, Jor03] but mainly consist on the same concept: updating prices until an equilibrium point is reached. The objective function to be optimized is the social welfare (a combination of network benefit and user surplus) under some capacity constraints (bandwidth and/or buffer) that guarantee the QoS level. The problem is decomposed into a network problem of updating prices and a user problem of choosing adequate resource demand given its price. In [Jor03], an arbitrager layer is introduced to remove direct consideration of network resource while focusing in the QoS parameters. Indeed, the arbitrager negotiates with users optimal amount of loss probability (for instance) and then purchases the needed bandwidth from the
network. In [CSMK00] an intelligent agent is introduced order to replace the user in choosing the willingness to pay based on user learned preferences.

Effective Bandwidth

Effective Bandwidth [Kelly 97] is an important concept to support QoS over IP. It corresponds to the notion of equivalent bandwidth in ATM. It maps the QoS required to satisfy a request into an amount bandwidth that can be used by the admission control rule. A simple linear tariff is proposed which is tangent to the bandwidth evolution curve. This is an incentive to the users to declare (the more precisely) their real requirements.

Auctions

In [EB03, EB04], authors propose a Distributed Multi-link Auction mechanism (referred by DiMA) that deals with request connection establishment in order to provide some End-to-end guarantees of services. The DiMA mechanism determines hop-by-hop the path to be taken by a request while reserving the required resource over it (Effective bandwidth can be used for this issue). It consists of consecutive local auctions that each request has to win in order to be satisfied. Auctions take place in different time intervals considering only new requests since bandwidth assigned to accepted requests is not renegotiated.

In [BR03], authors propose a mechanism combining optimal provisioning of Label Switched Paths and auction-based pricing. The provisioning problem is solved as a multi-commodity flow problem, then bandwidth is assigned to competing traffic classes with respect to a second-price auction at a more fast time-scale.

Discussion

We have presented some of the existing pricing mechanisms under the Best Effort paradigm and for the emerging guaranteed performance services. The question that arises is how to integrate both approaches in order to deal with networks providing both services. Only few papers address this problem as in [BBMM03]. A proportional fairness pricing is applied to best effort traffic, then different admission control rules are defined based on a comparison between the benefit from best effort and connection to be established. One point of interest is to investigate the possibility of integrating other pricing mechanism together to propose a unified and a realistic approach of pricing on heterogeneous networks.

references


[Cadenus] http://www.cadenus.org/Public Deliverables


[Li01] W. Li, "Overview of Fine Granularity Scalability in MPEG-4 Video Standard" IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, VOL. 11, NO. 3, March 2001


[LiLiu03] Li, Liu, “Multirate Video Multicast over the Internet – an overview”, IEEE Network 2003


[NS2] UCB/LBNL/VINT Network simulator – NS (version 2) http://www.isi.edu/nsnam/ns


[P.833] ITU-T Recommendation P.833, "Methodology for derivation of equipment impairment factors from subjective listening-only tests" (02/2001).


[ZhFe93] H. Zhang, D. Ferrari, „Rate-controlled Static-Priority Queueing”, proceedings of IEEE INFOCOM’93.

4. Open Problems

This chapter discusses some open problems that will be focused by research in the near future.

4.1. Bandwidth Sharing between Elastic Flows with Unresponsive Traffic

The future Internet will support a wide range of services (voice, video, data). This integration of services on a common infrastructure has several advantages from the operational point of view and in the supporting of future applications whose characteristics are unknown. However, joining heterogeneous services with different QoS imposes several challenging issues in order to provide useful services with end-to-end guarantees.

Most of the traffic in a multiservice network can be decomposed in flows which can be classified as stream or elastic. Stream flows are produced by audio and video applications and its intrinsic rate has to be preserved as the flow passes through the network. Elastic flows results from the transfer of digital documents and the total transfer delay constitutes the main performance parameter.

The current Internet traffic is dominated by elastic flows which use TCP. Most of these flows are very short, corresponding to the transfer of small files (e.g., web page retrieval) and their rate is constrained by TCP slow start phase. Such flows do not adapt to the level of congestion of the network and are referred to as non-responsive traffic. On the other hand, long flows, which result from the transfer of files of several Mbytes (e.g., MP3 tracks), depend more on TCP congestion avoidance which adjusts instantaneously the rate of the flows in response to the network conditions.

With the emerging of streaming flows produced by multimedia applications the amount of the unresponsive traffic on the Internet can increase significantly in the near future. This traffic is carried by the UDP protocol which does not adapt to the current network state.

A natural assumption and widely adopted for modelling the bandwidth sharing among elastic flows is the processor sharing (PS) discipline. There has been considerable work on the performance of bandwidth sharing between long elastic flows, see e.g., [Massoulié 99, Fredj 01, Boyer 03].

However, the problem of integration and the impact that unresponsive traffic has on long elastic flows has received until now little attention. From a modelling point of view, the unresponsive traffic can be viewed as if the transmission capacity available for long flows varies over time. In [Nunez Queija 98] the distribution on the number of customers in a
M/M/1 processor sharing where the service rate changes according to a birth and dead process has been investigated. Similar models have been studied in [Nunez Queija 01] for the case of quasi-birth and death process.

Most of the work on the literature has been conducted in the study of the integration of elastic and streaming flows. In [Borst 02] a generalized PS discipline is used for service differentiation and prioritization of the two flows. They have shown that the asymptotic transfer delay of the elastic traffic is not affected by the stream traffic when the weight of the GPS for elastic traffic is larger than its average traffic rate. In [key 03] it is assumed that streaming traffic shares the bandwidth fairly with the elastic traffic. Through the scaling of a Markov process which describes the number of various flows in the network, they establish the stability of the model and give insights on the impact of each flow type on the other. Stochastic bounds for the mean number of elastic flows have been established in [Delcoigne 04]. Streaming traffic was considered as a priority service and was represented by a semi-Markov process; the remaining available bandwidth was shared by the elastic traffic in PS fashion.

The impact of short flows on long flows has been studied recently in [Fricker 04]. By representing the aggregation of bandwidth of short flows as a Orstein-Uhlenbeck process and using perturbation analysis, they present an expansion for the stationary distribution of the number of long flows active when its service rate is weakly perturbated by the OU process.

### 4.2. Traffic Characterization

Some open issues in traffic characterization have been identified before in section 3.5.3.2. They are summarized below.

Little is understood about how the traffic from a single audio or video source is shaped due to multiplexing and switching through a series of nodes in the network. Clearly, a good understanding of the impact of traffic shaping through the network nodes is key to the design of robust and efficient admission control procedures.

To this end, it is important to develop good and realistic models for the aggregate of heterogeneous traffic sources and to understand the behavior of such aggregate traffic as it traverses (multiplexed and de-multiplexed) one or a series of network nodes.

The properties of the aggregated traffic depend on the traffic mix and the service differentiation in the routers. Audio and video traffic compete individually, as they also do with other traffic like TCP-traffic. Most current differentiation schemes make it difficult to
assess the end-to-end properties in an analytical way. To be able to do that, admission control and service differentiation -and even possibly traffic separation-, must be done a smart way. Work by Kortebi [KQR] suggests a solution in that direction.

As pointed out earlier, new audio and video solutions will generate new types of traffic; measurement data for these are missing, and new source models/new parameterization is needed. This, together with new ways of handling the traffic, requires new models for the aggregated traffic.

Although rigorous or approximate analysis of the above is sought, it may be very difficult, if at all possible. A feasible and promising alternative, however, is to gain insight and understanding of the above through extensive measurements, realistic traces and simulation studies and laboratory setups.

Evidence and possible causes of long range dependence (LRD) and self-similarity in single (and aggregate) video and Internet traffic sources have been reported and discussed extensively in recent literature [Leland94] [GW94] [Beran95] [PF95] [ENW96] [CB96] [Norros94]. Heavy-tail distributions characterizing file transfers have also been observed and the analysis of some simple models has been considered [Cohen98] [BD98] [BC99] [BBJ03]. While the impact of these sources on network traffic (and, hence, performance) is undisputable, there is still much to learn and understand about their aggregate behavior, their shaping through network nodes, their impact on each other and on other (conventional) traffic sources, etc.

Only a few traffic (performance) models involving self-similar arrival processes and/or heavy-tail service time distributions can be found in recent literature [Norros94] [Cohen98] [BD98] [BC99] [BBJ03]. Much more must be done to gain a better insight and understanding of these traffic sources. Needless to say, many of the questions raised above (for conventional traffic sources) are also, if not more, valid in the presence of self-similar traffic and heavy-tailed distributions. How important it is to model this LRD for traffic where short buffers are needed may be questionable.

4.3. Admission Control

Some open issues in admission control have been identified before in section 3.5.4.1. They are summarized below.

As noted before, accurate characterization of individual sources and aggregate traffic as they traverse network nodes is crucial for efficient resource management and robust admission
control. The traffic model of a given source (or aggregate traffic) at the ingress of the network is no longer valid characterization of the same traffic at internal network nodes. Because of this, model-based admission control algorithms may not be expected to perform favorably in real network environments.

Carefully designed measurement-based admission control algorithms, however, have a better chance of achieving `good' performance [SYT99]. This is because resource allocation and admission decisions are based on active on-line measurements of the aggregate traffic at network nodes along the path of the requested connection [GT99].

Still, however, an incoming request for admission must provide a traffic characterization of the associated source, as well as minimum quality of service requirement. While the provided characterization of the source is no longer valid at internal network nodes, the impact of this misrepresentation may be insignificant at nodes switching large volumes of aggregate traffic. But this is not the case at (or close to) the ingress of the network.

Novel and creative (model- or measurement-based) methodologies need to be developed in order to enhance the performance of admission control in this regard.

Given the lack of results and limited experience regarding the dynamic behavior of aggregate traffic and self-similar traffic, admission control based on measurements and the time-scale decomposition approach [GT03] may hold much promise and should be further explored. Another approach that deserves more attention is that based on aggregate traffic envelopes [QK01], as it caters for a wide range of traffic types.

Scalability and dependability issues favor distributed algorithms for network management and admission control [KKZ00]. Distributed, simple, and robust mechanisms are becoming increasingly essential for management and operation of future and large networks - a rich and less explored area of research.

4.4. **Inter Domain Admission Control**

Some open issues in Inter Domain Admission Control have been identified before in section 3.1.3. They are summarized below:

- Problem of traffic characterization
  
  The proposed measurement-based method of Traffic Descriptor re-calculation is only one of the possible approaches for assessment of the traffic deformation caused by passing a domain. Other approaches giving lower over-dimensioning of the Traffic Descriptor
should be considered. The impact of the effect of traffic profile deformation in the case of other network services should also be carefully investigated.

- Inter-working (mapping) between services offered in different domains

In the general case, QoS services offered by different domains do not have to be identical. The differences can be related with the level of offered QoS guarantees, the methods for traffic characterization, traffic handling and admission control. In order to provide consistent end-to-end QoS service, the rules for possible inter-working between these different services must be precisely defined.

- QoS splitting/assembling functions

The QoS requirements of a flow are expressed in terms of end-to-end parameters, thus they are not directly related with the QoS level offered within a single domain. Therefore, on each step on the end-to-end path, the Inter Domain Admission Control has to assess the QoS degradation level expected in that particular domain and inform the next domain what it has to offer e.g. lower delay and packet loss ratio, taking into account the degradation introduced in all previously visited domains. This task may be realized with the help of so-called QoS assembling functions, which compose the end-to-end QoS values based on the contributions of consecutive domains. This is also a quite complex problem due to the variety of QoS parameters and their features. For example, the maximum packet delay is additive while the packet loss ratio is rather multiplicative.

**references**


5. Conclusions

Our conclusions on QoS requirements and class of service differentiation are:

- overprovisioning of network resources is the simplest and most used QoS technique, although uneconomical due to the resulting waste of network resources;

- traffic may be classified in classes to differentiate the QoS obtained. The QoS of a class may be guaranteed, not guaranteed but better than the default best effort, best effort, or worse than best effort. An alternative quantitative schema makes the QoS of one class consistently proportional to the QoS of some other class.

Our conclusions on Queuing Mechanisms are:

- Generalized Processor Sharing (GPS) is the ideal work-conserving fair queuing mechanism, which can be approximated by several other queuing mechanisms;

- the choice should result from a trade-off between the following factors: ease of implementation, fairness and protection, performance bounds, ease and efficiency of admission control.

Our conclusions on pricing from the provider point of view are as follows:

- return on investment must be assured by appropriately sharing network capital and operating costs between users;

- price discrimination is economically efficient but should be based on criteria other than pretended QoS guarantees;

- congestion pricing, used to efficiently share a scarce resource, is not a satisfactory charging basis for a commercial network operator;

- user preference for simplicity and transparency can be satisfied by a simple volume-based charging scheme in a network equipped with admission control.

Our conclusions on Inter Domain Admission Control are:

- due to multiplexing with other traffic inside the domain, the characteristics of a traffic flow may change, affecting the support of end-to-end QoS in a multi-domain network;

- a method to estimate this traffic profile deformation was analyzed. It improves the effectiveness of admission control and, as a consequence, the end-to-end QoS offered to the user.
Admission Control for Service Level Agreement management is accomplished by:

- first choosing the links which will be used to carry the traffic, usually a tree to the possible destinations or sources;
- then, a measurement-based or parameter-based approach is used to decide if the new flow can be accepted.

The conclusions related to SLA/SLS Management in Multi-domain Environment are:

- a number of concepts/features are common to all projects discussed which illustrate in fact what is the state of the art in the domain: they target to End-to-End QoS control with several classes of services and levels of guarantees; achieve QoS aware service in large network configurations; are based on PBNM concepts implemented in a distributed way; implement SLA/SLSs between customers and providers; use AC at service invocation epochs; use resource management and traffic engineering (TE) in order to optimise the utilization of resources; use a monitoring system to provide feedback on network and resource status to the management function; are supported by DiffServ/MPLS technologies for traffic control.

The conclusions on traffic prediction are:

- "classical" and chaotic-based predictor can be used for obtaining information on the future behaviour of network traffic;
- chaotic predictors have the advantage of taking into account the self-similarity of network traffic.

Our conclusions on service differentiation for audio and video streams are:

- some techniques should be used to maximize the perceived quality for audio and video: coding techniques that compress the data, error resilience techniques that assure graceful degradation in the (inevitable) presence of packet loss and transmission errors;
- traffic characterization plays an important role for service guarantee and admission control;
- admission control with proper resource reservation can ensure that a newly admitted flow can receive its required service guarantees.
# List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABE</td>
<td>Alternative Best Effort</td>
</tr>
<tr>
<td>AC</td>
<td>Admission Control</td>
</tr>
<tr>
<td>ACA</td>
<td>Admission Control Agent</td>
</tr>
<tr>
<td>AD</td>
<td>Administrative Domain</td>
</tr>
<tr>
<td>AF</td>
<td>Assured Forwarding</td>
</tr>
<tr>
<td>AM</td>
<td>Access Mediator</td>
</tr>
<tr>
<td>AS</td>
<td>Autonomous Systems</td>
</tr>
<tr>
<td>ATM</td>
<td>Asynchronous Transfer Mode</td>
</tr>
<tr>
<td>AWTP</td>
<td>Advanced Waiting Time Priority</td>
</tr>
<tr>
<td>BB</td>
<td>Bandwidth Broker</td>
</tr>
<tr>
<td>BE</td>
<td>Best Effort</td>
</tr>
<tr>
<td>BER</td>
<td>Bit Error Ratio</td>
</tr>
<tr>
<td>BGP</td>
<td>Border Gateway Protocol</td>
</tr>
<tr>
<td>BGRP</td>
<td>Border Gateway Resource Reservation Protocol</td>
</tr>
<tr>
<td>bps</td>
<td>bit per second</td>
</tr>
<tr>
<td>CAC</td>
<td>Connection Admission Control</td>
</tr>
<tr>
<td>CBR</td>
<td>Constant Bit Rate</td>
</tr>
<tr>
<td>CBWFQ</td>
<td>Class Based Weighted Fair Queuing</td>
</tr>
<tr>
<td>CDVT</td>
<td>Cell Delay Variation Tolerance</td>
</tr>
<tr>
<td>CPC</td>
<td>Central Policy Controller</td>
</tr>
<tr>
<td>cSLA</td>
<td>Customer Service Level Agreement</td>
</tr>
<tr>
<td>cSLS</td>
<td>Customer Service Level Specification</td>
</tr>
<tr>
<td>D-BIND</td>
<td>Deterministic Bounding Interval Dependent</td>
</tr>
<tr>
<td>DCT</td>
<td>Discrete Cosine Transform</td>
</tr>
<tr>
<td>DiffServ</td>
<td>Differentiated Services</td>
</tr>
<tr>
<td>DiMA</td>
<td>Distributed Multi-link Auction mechanism</td>
</tr>
<tr>
<td>DRP</td>
<td>Dynamic Resource Pool</td>
</tr>
<tr>
<td>DRR</td>
<td>Deficit Round Robin</td>
</tr>
<tr>
<td>DRTM</td>
<td>Dynamic Route Management</td>
</tr>
<tr>
<td>DSCP</td>
<td>Differentiated Services Codepoint</td>
</tr>
<tr>
<td>DSD</td>
<td>Duplicate Scheduling with Deadlines</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>E2E</td>
<td>end-to-end</td>
</tr>
<tr>
<td>EAT</td>
<td>End-User Application Toolkit</td>
</tr>
<tr>
<td>EBB</td>
<td>Exponentially Bounded Burstiness</td>
</tr>
<tr>
<td>ECN</td>
<td>Explicit Congestion Notification</td>
</tr>
<tr>
<td>EDD</td>
<td>Earliest Due Date</td>
</tr>
<tr>
<td>EF</td>
<td>Expedited Forwarding</td>
</tr>
<tr>
<td>e-QC</td>
<td>extended QoS Class</td>
</tr>
<tr>
<td>FCFS</td>
<td>First Come First Served</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>FEC</td>
<td>Forward Error Correction</td>
</tr>
<tr>
<td>FGS</td>
<td>Fine Granular Scalability</td>
</tr>
<tr>
<td>FIFO</td>
<td>First In First Out</td>
</tr>
<tr>
<td>FP5</td>
<td>Fifth Framework Programme</td>
</tr>
<tr>
<td>FP6</td>
<td>Sixth Framework Programme</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>GCRA</td>
<td>Generic Cell Rate Algorithm</td>
</tr>
<tr>
<td>GOP</td>
<td>Group Of Pictures</td>
</tr>
<tr>
<td>GPS</td>
<td>Generalized Processor Sharing</td>
</tr>
<tr>
<td>gSBB</td>
<td>generalized SBB</td>
</tr>
<tr>
<td>HDTV</td>
<td>High-Definition Television</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hypertext Transfer Protocol</td>
</tr>
<tr>
<td>IDAC</td>
<td>Inter-Domain Admission Control</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>IntServ</td>
<td>Integrated Services</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPDV</td>
<td>IP Packet Delay Variation</td>
</tr>
<tr>
<td>IPTD</td>
<td>IP Packet Transfer Delay</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>IST</td>
<td>Information Society Technologies</td>
</tr>
<tr>
<td>ITU</td>
<td>International Telecommunications Union</td>
</tr>
<tr>
<td>ITU-T</td>
<td>International Telecommunications Union Telecommunication Standardization Sector</td>
</tr>
<tr>
<td>LDAP</td>
<td>Lightweight Directory Access Protocol</td>
</tr>
<tr>
<td>LMK</td>
<td>Least Mean Kurtosis</td>
</tr>
<tr>
<td>LMS</td>
<td>Least Means Square</td>
</tr>
<tr>
<td>LMMSE</td>
<td>Linear Minimum Mean Square Error</td>
</tr>
<tr>
<td>LRD</td>
<td>Long Range Dependence</td>
</tr>
<tr>
<td>MB</td>
<td>Macro-Block</td>
</tr>
<tr>
<td>MB</td>
<td>Measurement-based</td>
</tr>
<tr>
<td>MBAC</td>
<td>Measurement-Based Admission Control</td>
</tr>
<tr>
<td>MD</td>
<td>Multiple Description</td>
</tr>
<tr>
<td>MDRR</td>
<td>Modified Deficit Round Robin</td>
</tr>
<tr>
<td>MMSE</td>
<td>Minimum Mean Square Error</td>
</tr>
<tr>
<td>MOS</td>
<td>Mean Opinion Score</td>
</tr>
<tr>
<td>MPEG</td>
<td>Motion Picture Experts Group</td>
</tr>
<tr>
<td>MPLS</td>
<td>Multiprotocol Label Switching</td>
</tr>
<tr>
<td>MTU</td>
<td>Maximum Transfer Unit</td>
</tr>
<tr>
<td>ND</td>
<td>Network Dimensioning</td>
</tr>
<tr>
<td>NDA</td>
<td>Non-Disclosure Agreement</td>
</tr>
<tr>
<td>NLMS</td>
<td>Normalized Least Means Square</td>
</tr>
<tr>
<td>NP</td>
<td>Network Provider</td>
</tr>
<tr>
<td>NP/NO</td>
<td>Network Provider/Operator</td>
</tr>
<tr>
<td>NS-2</td>
<td>Network Simulator version 2</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>NTSC</td>
<td>National Television System Committee</td>
</tr>
<tr>
<td>OA</td>
<td>Ordered Aggregate</td>
</tr>
<tr>
<td>ODP</td>
<td>Open Distributed Processing</td>
</tr>
<tr>
<td>o-QC</td>
<td>offered QoS Class</td>
</tr>
<tr>
<td>PAL</td>
<td>Phase Alternation Line</td>
</tr>
<tr>
<td>PAPI</td>
<td>Policy Application Programming Interface</td>
</tr>
<tr>
<td>PB</td>
<td>Parameter-based</td>
</tr>
<tr>
<td>PBC</td>
<td>Probabilistic Burstiness Curve</td>
</tr>
<tr>
<td>PBNM</td>
<td>Policy Based Network Management</td>
</tr>
<tr>
<td>PDB</td>
<td>Per Domain Behaviour</td>
</tr>
<tr>
<td>PDP</td>
<td>Policy Decision Point</td>
</tr>
<tr>
<td>PEP</td>
<td>Policy Enforcement Points</td>
</tr>
<tr>
<td>PGPS</td>
<td>Packet-by-packet Generalized Processor Sharing</td>
</tr>
<tr>
<td>PHB</td>
<td>Per-Hop Behavior</td>
</tr>
<tr>
<td>PHYP</td>
<td>Physical Connectivity Provider</td>
</tr>
<tr>
<td>PIN</td>
<td>Policy Ignorant Nodes</td>
</tr>
<tr>
<td>PLR</td>
<td>Packet Loss Ratio</td>
</tr>
<tr>
<td>PMP</td>
<td>Paris Metro Pricing</td>
</tr>
<tr>
<td>PN</td>
<td>Private Networks</td>
</tr>
<tr>
<td>PQ</td>
<td>Priority Queuing</td>
</tr>
<tr>
<td>PQWFQ</td>
<td>Priority Queuing Weighted Fair Queuing</td>
</tr>
<tr>
<td>PS</td>
<td>Policy Servers</td>
</tr>
<tr>
<td>PS</td>
<td>Processor Sharing</td>
</tr>
<tr>
<td>pSLA</td>
<td>Provider Service Level Agreement</td>
</tr>
<tr>
<td>pSLS</td>
<td>Provider Service Level Specification</td>
</tr>
<tr>
<td>PSP</td>
<td>Progressive Second Price</td>
</tr>
<tr>
<td>PS-QMAR</td>
<td>Priority Scheme based on Queue Management Algorithm with Reservations</td>
</tr>
<tr>
<td>QC</td>
<td>QoS Class</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAM</td>
<td>Resource Availability Matrix</td>
</tr>
<tr>
<td>RC</td>
<td>Resource Controller</td>
</tr>
<tr>
<td>RCL</td>
<td>Resource Control Layer</td>
</tr>
<tr>
<td>RCSP</td>
<td>Rate-Controlled Static-Priority</td>
</tr>
<tr>
<td>RED</td>
<td>Random Early Detection</td>
</tr>
<tr>
<td>RFC</td>
<td>Request For Comments</td>
</tr>
<tr>
<td>RM</td>
<td>Resource mediator</td>
</tr>
<tr>
<td>RPC</td>
<td>Resource Provisioning Cycle</td>
</tr>
<tr>
<td>RR</td>
<td>Round Robin</td>
</tr>
<tr>
<td>RS</td>
<td>Reed-Solomon</td>
</tr>
<tr>
<td>RSVP</td>
<td>Resource ReSerVation Protocol</td>
</tr>
<tr>
<td>RVLC</td>
<td>Reversible Variable Length Coding</td>
</tr>
<tr>
<td>S-BIND</td>
<td>Stochastic Bounding Interval Dependent</td>
</tr>
<tr>
<td>SBB</td>
<td>Stochastically Bounded Burstiness</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>SCFQ</td>
<td>Self-Clocked Fair Queuing</td>
</tr>
<tr>
<td>SDTV</td>
<td>Standard-Definition Television</td>
</tr>
<tr>
<td>SIC</td>
<td>Sensitive dependence on Initial Conditions</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SLS</td>
<td>Service Level Specification</td>
</tr>
<tr>
<td>SM</td>
<td>Service Mediator</td>
</tr>
<tr>
<td>SMPTE</td>
<td>Society of Motion Picture and Television Engineers</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal-to-Noise Ratio</td>
</tr>
<tr>
<td>SP</td>
<td>Service Provider</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>SrNP</td>
<td>Service Negotiation Protocol</td>
</tr>
<tr>
<td>STFQ</td>
<td>Start-Time Fair Queuing</td>
</tr>
<tr>
<td>TCB</td>
<td>Traffic Conditioning Blocks</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TD</td>
<td>Traffic Descriptor</td>
</tr>
<tr>
<td>TE</td>
<td>Traffic Engineering</td>
</tr>
<tr>
<td>TT</td>
<td>Traffic Trunk</td>
</tr>
<tr>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
<tr>
<td>UMTS</td>
<td>Universal Mobile Telecommunications System</td>
</tr>
<tr>
<td>VBR</td>
<td>Variable Bit Rate</td>
</tr>
<tr>
<td>VLC</td>
<td>Variable Length Coding</td>
</tr>
<tr>
<td>VoIP</td>
<td>Voice over Internet Protocol</td>
</tr>
<tr>
<td>VPN</td>
<td>Virtual Private Network</td>
</tr>
<tr>
<td>WBB</td>
<td>Weibull Bounded Burstiness</td>
</tr>
<tr>
<td>WEDD</td>
<td>Weighted Earliest Due Date</td>
</tr>
<tr>
<td>WFQ</td>
<td>Weighted Fair Queuing</td>
</tr>
<tr>
<td>WRR</td>
<td>Weighted Round Robin</td>
</tr>
<tr>
<td>WTP</td>
<td>Waiting Time Priority</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>