



INSTITUTO  
SUPERIOR  
TÉCNICO

**UNIVERSIDADE TÉCNICA DE LISBOA**  
**INSTITUTO SUPERIOR TÉCNICO**

**POLÍTICAS ACTIVAS**  
**PARA GESTÃO DE REDES**

Paulo Rogério Barreiros d'Almeida Pereira  
(Mestre)

**Dissertação para obtenção do grau de Doutor em**  
**Engenharia Electrotécnica e de Computadores**

Orientador: Doutor Paulo da Costa Luís da Fonseca Pinto  
Co-Orientador: Doutor Vítor Manuel Paulino Vargas  
Co-Orientador: Doutor Djamel Fawzi Hadj Sadok

JÚRI

Presidente: Reitor da Universidade Técnica de Lisboa  
Vogais: Doutor Augusto Júlio Domingues Casaca  
Doutor Fernando Pedro Lopes Boavida Fernandes  
Doutor Paulo da Costa Luís da Fonseca Pinto  
Doutor Carlos Alberto de Carvalho Belo  
Doutor Vítor Manuel Paulino Vargas  
Doutora Teresa Maria Sá Ferreira Vazão Vasques

**Janeiro de 2003**



**Aos meus pais**



# RESUMO

A concorrência tem levado a gestão de redes a precisar de incluir também a gestão dos serviços e dos processos de negócio das empresas. Pretende-se, desta forma, avaliar o nível de qualidade dos serviços oferecido, maximizando-o, e se possível assegurando-o.

Para tratar este problema, propõe-se a utilização de um modelo de gestão baseado numa hierarquia de políticas activas, com vários níveis de abstracção, para gerir a rede e os sistemas envolvidos, avaliar a qualidade de serviço oferecida, e na medida do possível, assegurar os requisitos de qualidade pretendidos. As políticas activas são objectos activos que actuam sobre a rede, automatizando as tarefas de gestão, cobrindo monitorização, controlo e planeamento, com vista a atingir os objectivos de gestão propostos.

Propõe-se ainda uma metodologia para analisar os requisitos de qualidade de serviço, e construir as políticas activas necessárias com base num conjunto de regras de actuação predefinidas que ajustam dinamicamente os parâmetros dos objectos geridos.

Foram simulados dois cenários para ilustrar a aplicação da metodologia e a arquitectura de gestão propostas. Mostrou-se, assim, como o modelo proposto pode ser aplicado para analisar problemas de gestão, e como se podem utilizar políticas activas para fazer cumprir acordos de níveis de serviço entre fornecedores de serviço e clientes.



# ABSTRACT

Competition has raised the need for network management to also include companies' service and business process management. The purpose is to rate the quality level of the services, maximising it, and if possible assuring it.

To address these issues, this thesis proposes a management model based on an active policy hierarchy, with several abstraction levels, to manage the network and the associated systems, rate the offered quality of service, and as far as possible assure the expected quality requirements. Active policies are active objects that act over the network, automating management tasks, dealing with monitoring, control and planning, in order to reach the proposed management objectives.

This thesis also proposes a methodology to analyse the quality of service requirements, and to build the necessary active policies based on a pre-defined set of acting rules that dynamically adjust managed object parameters.

Two scenarios were simulated to show how the proposed methodology and management architecture should be applied. Thus, it was shown how the proposed model can be applied to analyse management problems, and how active policies can be used to enforce service level agreements between service providers and users.





# **PALAVRAS-CHAVE**

Gestão de Redes.

Gestão de Sistemas Distribuídos.

Políticas Activas.

Políticas de Gestão.

Gestão de Níveis de Serviço.

Qualidade de Serviço.

# **KEYWORDS**

Network Management.

Distributed Systems Management.

Active Policies.

Management Policy.

Service Level Management.

Quality of Service.



# AGRADECIMENTOS

A realização deste trabalho só se tornou possível graças ao apoio de professores, colegas, amigos e familiares.

Ao meu orientador científico, Prof. Paulo Pinto, agradeço a sua amizade, o seu apoio, os seus ensinamentos e o encorajamento.

Ao Prof. Djamel Sadok, co-orientador científico, agradeço a sua amizade, e ter-me introduzido aos serviços diferenciados e ao simulador *ns*.

Aos restantes membros do Departamento de Informática, agora Centro de Informática, da Universidade Federal de Pernambuco, no Recife - Brasil, agradeço a troca de ideias, o convívio e apoio durante as estadias no Brasil. Devo referir, em especial, a Prof.<sup>a</sup> Judith Kelner e vários alunos de pós-graduação: os Drs. Elionildo Menezes, Carlos Kamienski, Joseane Farias, Carlos Cordeiro e Nelson Rosa.

Ao Prof. Vitor Vargas, co-orientador científico, agradeço a sua amizade, e a aceitação da co-orientação, em Setembro de 1998, após o Prof. Paulo Pinto ter mudado para a Universidade Nova de Lisboa.

Aos meus colegas, Eng.<sup>o</sup> Luís Bernardo e Eng.<sup>o</sup> José António Gaspar, já doutorados, agradeço a profunda amizade, permanente disponibilidade para discutir ideias e encorajamento.

Ao Prof. Paulo Veríssimo, Prof. Boavida Fernandes e Eng.<sup>o</sup> Luís Alveirinho, agradeço a disponibilidade para trocar ideias na fase inicial do doutoramento.

A amizade, a ajuda prestada e o interesse manifestado pelo meu trabalho por parte de amigos, colegas do Instituto Superior Técnico (IST) e do Instituto de Engenharia de Sistemas e Computadores (INESC) são também objecto da minha gratidão. Devo referir, em especial, o Prof. Augusto Casaca, a Prof.<sup>a</sup> Teresa Vazão, o Prof. Rui Crespo, o Prof. Rui Rocha, o Prof. Cunha Serra, o Prof. João Sobrinho, o Prof. Carlos Belo, o Prof. Sena da Silva, o Prof. Nelson Antunes, o Prof. Nuno Horta, a Eng.<sup>a</sup> Gracinda Carvalho, a Eng.<sup>a</sup> Filipa Ferreira, a Eng.<sup>a</sup> Isabel Faria, o Eng.<sup>o</sup> José Santiago, o Eng.<sup>o</sup> Miguel Correia, o Eng.<sup>o</sup> Paulo Sardinha, a Dr.<sup>a</sup> Sara Cavaco, o Jaime Pinto, o João Pedro Afonso, o Lu, a Min-Chun, o Coro de Santa Maria e a Ju.

Aos alunos Jorge Sousa, Pedro Teixeira, Luís Agostinho, Pedro Pessoa, Bruno Afonso e Daniel Gomes, agradeço terem trabalhado em trabalhos finais de curso na área de gestão de redes, manifestando interesse pela área desta tese.

Ao Instituto Superior Técnico (IST), agradeço o apoio institucional e os três anos de dispensa de serviço docente concedidos.

Ao Instituto de Engenharia de Sistemas e Computadores (INESC), agradeço todo o apoio institucional e computacional dado.

À Fundação para a Ciência e Tecnologia, agradeço o financiamento parcial do equipamento computacional, bibliografia, e participação em conferências, no âmbito de um projecto PRAXIS XXI, sob o contrato 2/2.1/TIT/1633/95.

Ao Instituto de Cooperação Científica e Tecnológica Internacional, agradeço o financiamento das missões ao Brasil ao abrigo do convénio ICCTI/CAPES - 99 - Proc° 423/CAPES.

Por fim, mas certamente o maior agradecimento é devido à minha família, especialmente aos meus pais e ao meu irmão, pelo apoio e incentivo que sempre me deram ao longo de toda a minha vida.

Lisboa, Janeiro de 2003

*Paulo Rogério Barreiros d'Almeida Pereira*

# ÍNDICE

|   |             |
|---|-------------|
| <b>RESUMO</b> .....                                 | <b>V</b>    |
| <b>ABSTRACT</b> .....                               | <b>VII</b>  |
| <b>PALAVRAS-CHAVE</b> .....                         | <b>IX</b>   |
| <b>KEYWORDS</b> .....                               | <b>IX</b>   |
| <b>AGRADECIMENTOS</b> .....                         | <b>XI</b>   |
| <b>ÍNDICE</b> .....                                 | <b>XIII</b> |
| <b>LISTA DE FIGURAS</b> .....                       | <b>XIX</b>  |
| <b>LISTA DE TABELAS</b> .....                       | <b>XXV</b>  |
| <b>CAPÍTULO 1 INTRODUÇÃO</b> .....                  | <b>1</b>    |
| 1.1 O PROBLEMA.....                                 | 1           |
| 1.2 ENQUADRAMENTO .....                             | 2           |
| 1.3 OBJECTIVOS E CONTRIBUIÇÕES.....                 | 6           |
| 1.4 ESTRUTURA DA TESE.....                          | 7           |
| 1.5 NOTAÇÕES UTILIZADAS.....                        | 8           |
| <b>CAPÍTULO 2 TRABALHO RELACIONADO</b> .....        | <b>9</b>    |
| 2.1 INTRODUÇÃO À GESTÃO DE REDES .....              | 9           |
| 2.2 MODELO FUNCIONAL DE GESTÃO.....                 | 11          |
| 2.2.1 Monitorização, Controlo e Planeamento .....   | 11          |
| 2.2.2 Áreas Funcionais de Gestão .....              | 13          |
| 2.2.3 Funções de Gestão de Sistemas .....           | 15          |
| 2.2.3.1 SMF de Atributos e Objectos de Medida ..... | 15          |
| 2.2.3.2 SMF de Domínios e Políticas de Gestão ..... | 16          |
| 2.2.3.3 SMF de Sequenciamento de Comandos .....     | 18          |
| 2.2.3.4 Discussão .....                             | 18          |

|   |           |
|---|-----------|
| 2.3 MODELO ARQUITECTURAL DE GESTÃO .....                | 18        |
| 2.3.1 Arquitecturas Centralizadas.....                  | 18        |
| 2.3.2 Arquitecturas Distribuídas.....                   | 20        |
| 2.4 MODELO DE INFORMAÇÃO DE GESTÃO.....                 | 24        |
| 2.5 MODELO RELACIONAL DE GESTÃO .....                   | 26        |
| 2.5.1 Protocolos de Gestão OSI e Internet.....          | 26        |
| 2.5.2 Gestão usando Objectos Distribuídos .....         | 28        |
| 2.5.3 Outros Protocolos de Gestão.....                  | 29        |
| 2.5.4 Segurança .....                                   | 30        |
| 2.5.5 Discussão .....                                   | 30        |
| 2.6 QUALIDADE DE SERVIÇO.....                           | 31        |
| 2.7 GESTÃO DE NÍVEIS DE SERVIÇO (SLM) .....             | 36        |
| 2.8 POLÍTICAS DE GESTÃO .....                           | 43        |
| 2.8.1 Aplicações das Políticas .....                    | 44        |
| 2.8.2 Classificação, Hierarquia e Refinamento .....     | 44        |
| 2.8.3 Conflitos entre Políticas .....                   | 47        |
| 2.8.4 Exemplo de Arquitectura de Políticas.....         | 49        |
| 2.8.5 Discussão .....                                   | 50        |
| 2.9 LINGUAGENS DE GESTÃO .....                          | 51        |
| 2.10 DISCUSSÃO E MOTIVAÇÃO PARA A TESE .....            | 52        |
| <b>CAPÍTULO 3   MODELO DE GESTÃO .....</b>              | <b>55</b> |
| 3.1 DESCRIÇÃO GERAL.....                                | 55        |
| 3.2 NÍVEIS DE ABSTRAÇÃO, HIERARQUIA E REFINAMENTO ..... | 56        |
| 3.3 POLÍTICAS ACTIVAS.....                              | 59        |
| 3.4 CONTRATOS .....                                     | 61        |
| 3.5 METODOLOGIA.....                                    | 63        |
| 3.5.1 Comparação de Sistemas Periciais .....            | 65        |
| 3.5.2 Funcionamento Interno das Políticas Activas ..... | 69        |

---

|  |           |
|--|-----------|
| 3.5.3 Construção do Comportamento das Políticas Activas.....   | 74        |
| 3.5.4 Critérios de Avaliação.....                              | 77        |
| 3.6 REGRAS DE ACTUAÇÃO PREDEFINIDAS.....                       | 78        |
| 3.7 ARQUITECTURA DE GESTÃO .....                               | 81        |
| 3.8 BENEFÍCIOS DAS POLÍTICAS ACTIVAS .....                     | 83        |
| <b>CAPÍTULO 4 ESTUDO DE UM CASO - REDE EMPRESARIAL.....</b>    | <b>85</b> |
| 4.1 CENÁRIO .....  | 85        |
| 4.2 APLICAÇÃO DO MODELO - ARQUITECTURA.....                    | 86        |
| 4.2.1 Nível Processo de Negócio .....                          | 86        |
| 4.2.2 Nível Serviço .....                                      | 86        |
| 4.2.3 Nível Tarefa.....  | 86        |
| 4.2.4 Nível Rede e Sub-Rede.....                               | 88        |
| 4.2.5 Nível Equipamento .....                                  | 89        |
| 4.3 SIMULAÇÃO.....   | 90        |
| 4.3.1 O Simulador <i>Ptolemy</i> .....                         | 90        |
| 4.3.2 Topologia Cliente / Servidor .....                       | 92        |
| 4.3.3 Parâmetros de Simulação .....                            | 95        |
| 4.3.4 Discussão .....  | 97        |
| 4.4 APLICAÇÃO DO MODELO - POLÍTICAS ACTIVAS .....              | 99        |
| 4.4.1 Política Activa de Gestão do Atraso das Vendas .....     | 101       |
| 4.4.2 Política Activa de Gestão de Controlo de Acessos .....   | 101       |
| 4.4.3 Política Activa de Gestão do Atraso da Aplicação .....   | 102       |
| 4.4.4 Política Activa de Gestão do Atraso da Base de Dados ... | 102       |
| 4.4.5 Política Activa de Gestão de <i>Backups</i> .....        | 103       |
| 4.4.6 Política Activa de Gestão de <i>Caches</i> .....         | 103       |
| 4.5 RESULTADOS OBTIDOS .....                                   | 104       |
| 4.6 AVALIAÇÃO .....  | 106       |
| 4.7 CONCLUSÕES.....  | 108       |

## **CAPÍTULO 5 ESTUDO DE UM CASO - FORNECEDOR DE SERVIÇO INTERNET .. 109**

|   |     |
|---|-----|
| 5.1 CENÁRIO .....   | 109 |
| 5.1.1 Introdução ao Problema.....   | 109 |
| 5.1.2 Introdução aos Serviços Diferenciados .....                         | 112 |
| 5.1.3 Discussão .....   | 117 |
| 5.2 APLICAÇÃO DO MODELO - ARQUITECTURA .....                              | 118 |
| 5.2.1 Nível Processo de Negócio.....                                      | 120 |
| 5.2.2 Nível Serviço.....  | 120 |
| 5.2.3 Nível Rede e Sub-Rede .....   | 122 |
| 5.2.3.1 Gestão da Admissão de Novas Ligações .....                        | 122 |
| 5.2.3.2 Gestão da Largura de Banda de Linhas .....                        | 125 |
| 5.2.3.3 Gestão de Falhas de Linhas .....                                  | 126 |
| 5.2.4 Nível Equipamento.....  | 127 |
| 5.2.4.1 Gestão da Largura de Banda por Classe de Tráfego .....            | 127 |
| 5.2.4.2 Gestão do Descarte de Pacotes .....                               | 129 |
| 5.3 SIMULAÇÃO .....   | 134 |
| 5.3.1 O <i>Network Simulator</i> .....                                    | 134 |
| 5.3.2 Topologia da Rede .....   | 136 |
| 5.3.3 Parâmetros de Simulação.....  | 138 |
| 5.3.4 Discussão .....   | 143 |
| 5.4 APLICAÇÃO DO MODELO - POLÍTICAS ACTIVAS .....                         | 146 |
| 5.4.1 Política Activa de Gestão dos Pesos das Classes de<br>Tráfego ..... | 148 |
| 5.4.2 Política Activa de Gestão de Largura de Banda das<br>Linhas.....    | 153 |
| 5.4.3 Política Activa de Gestão de Admissão de Ligações.....              | 163 |
| 5.4.4 Política Activa de Gestão de Linhas Extra.....                      | 169 |
| 5.4.5 Política Activa de Gestão de SLAs .....                             | 175 |
| 5.4.6 Política Activa de Gestão do Mapeamento para PHBs .....             | 176 |
| 5.5 FUNCIONAMENTO SIMULTÂNEO DAS VÁRIAS POLÍTICAS ACTIVAS.....            | 182 |



---

|   |            |
|---|------------|
| 5.5.1 Sem Política Activa de Planeamento .....                        | 182        |
| 5.5.2 Política Activa de Planeamento.....                             | 187        |
| 5.6 AVALIAÇÃO .....   | 192        |
| 5.7 CONCLUSÕES.....   | 194        |
| <b>CAPÍTULO 6 AVALIAÇÃO.....</b>                                      | <b>195</b> |
| 6.1 COMPARAÇÃO DO MODELO .....  | 195        |
| 6.2 COMPARAÇÃO DOS CENÁRIOS .....                                     | 197        |
| 6.3 CONCLUSÕES.....   | 199        |
| <b>CAPÍTULO 7 CONCLUSÃO .....</b>                                     | <b>201</b> |
| 7.1 SÍNTESE .....   | 201        |
| 7.2 CONCLUSÕES.....   | 202        |
| 7.3 TRABALHO FUTURO .....   | 203        |
| 7.3.1 Áreas de Investigação Futura.....                               | 203        |
| 7.3.2 Melhoramentos na Aplicação Prática do Modelo .....              | 205        |
| <b>BIBLIOGRAFIA .....</b>   | <b>207</b> |
| <b>APÊNDICE A LISTA DE SIGLAS.....</b>                                | <b>225</b> |
| <b>APÊNDICE B INTRODUÇÃO AOS CONTROLADORES PID.....</b>               | <b>231</b> |
| <b>APÊNDICE C ANÁLISE ESTATÍSTICA DE RESULTADOS DE SIMULAÇÃO.....</b> | <b>235</b> |



# LISTA DE FIGURAS

|   |    |
|---|----|
| Figura 1.1: Camadas da arquitectura TMN.....                                  | 2  |
| Figura 1.2: Relações entre os diferentes tipos de gestão.....                 | 5  |
| Figura 2.1: Operações de monitorização, controlo e planeamento.....           | 11 |
| Figura 2.2: Horizontes temporais e actividades de gestão.....                 | 12 |
| Figura 2.3: Exemplo de limiares da SMF de atributos e objectos de medida..... | 16 |
| Figura 2.4: Exemplo de domínios e políticas associadas.....                   | 17 |
| Figura 2.5: Arquitectura gestor / agente.....                                 | 19 |
| Figura 2.6: Métricas para avaliar benefícios da descentralização.....         | 20 |
| Figura 2.7: Arquitectura hierárquica de gestão.....                           | 22 |
| Figura 2.8: Arquitectura de gestão baseada em agentes móveis.....             | 22 |
| Figura 2.9: Exemplo de MIB Internet.....                                      | 25 |
| Figura 2.10: Exemplo de descrição de objecto gerido na SMI Internet.....      | 25 |
| Figura 2.11: Pilhas de protocolos na gestão OSI e Internet.....               | 27 |
| Figura 2.12: Operação do protocolo de gestão OSI para a primitiva GET.....    | 28 |
| Figura 2.13: Limites e limiares de QoS.....                                   | 35 |
| Figura 2.14: Conceitos envolvidos em SLM.....                                 | 37 |
| Figura 2.15: Ciclo de revisão de um SLA.....                                  | 39 |
| Figura 2.16: Ciclo de vida de um serviço e do seu SLA.....                    | 40 |
| Figura 2.17: Exemplo de gestão de falhas numa arquitectura SLM.....           | 41 |
| Figura 2.18: Critérios para classificação de políticas.....                   | 45 |
| Figura 2.19: Arquitectura hierárquica de políticas.....                       | 46 |
| Figura 2.20: Classificação de conflitos entre políticas.....                  | 48 |
| Figura 2.21: Arquitectura de controlo de admissão baseado em políticas.....   | 49 |
| Figura 2.22: Classe que define uma regra de uma política.....                 | 50 |
| Figura 3.1: Exemplo de níveis de abstracção.....                              | 57 |
| Figura 3.2: Exemplo de refinamento de requisitos de uma venda.....            | 58 |
| Figura 3.3: Estrutura interna das políticas activas.....                      | 60 |
| Figura 3.4: Conceitos difusos de carga “baixa”, “média” e “alta”.....         | 66 |

|   |     |
|---|-----|
| Figura 3.5: Neurónio artificial de uma rede neuronal. ....  | 67  |
| Figura 3.6: Bloco de monitorização de uma política activa. ....   | 70  |
| Figura 3.7: Bloco de gestão reactiva de uma política activa. ....   | 73  |
| Figura 3.8: Bloco de gestão proactiva de uma política activa. ....  | 74  |
| Figura 3.9: Modelo de um parâmetro de QoS de gestão. ....   | 75  |
| Figura 3.10: Algoritmo genérico de ajuste de parâmetros. ....   | 76  |
| Figura 3.11: Avaliação da convergência para um objectivo. ....  | 78  |
| Figura 3.12: Funcionamento da regra básica de limiares com histerese. ....  | 78  |
| Figura 3.13: Funcionamento da regra <code>adjust_increment</code> . ....  | 80  |
| Figura 3.14: Exemplo de utilização da regra <code>pid_control</code> . ....   | 81  |
| Figura 3.15: Arquitectura para implementar a hierarquia de políticas activas. ....  | 82  |
| Figura 4.1: Refinamento do requisito do tempo de processamento de uma<br>venda. ....  | 87  |
| Figura 4.2: Contribuições para o atraso médio na base de dados. ....  | 88  |
| Figura 4.3: Contribuições para a carga na base de dados. ....   | 88  |
| Figura 4.4: Contribuições para o atraso médio obtido pelos clientes. ....   | 89  |
| Figura 4.5: Topologia simulada representando os clientes e servidores. ....   | 92  |
| Figura 4.6: Topologia de um cliente. ....   | 93  |
| Figura 4.7: Topologia de um servidor. ....  | 93  |
| Figura 4.8: Topologia do bloco medidor de atrasos. ....   | 94  |
| Figura 4.9: Tráfego médio horário num segmento de rede local no período de<br>uma semana. ....  | 96  |
| Figura 4.10: Número médio de acessos em cada hora ao servidor<br><code>www.ist.utl.pt</code> . ....                                       | 97  |
| Figura 4.11: Políticas activas utilizadas e as suas relações. ....  | 100 |
| Figura 4.12: Comparação do atraso médio de uma venda em função da carga de<br>pedidos nas situações com e sem as políticas activas. ....  | 105 |
| Figura 4.13: Comparação do atraso máximo de uma venda em função da carga<br>de pedidos nas situações com e sem as políticas activas. .... | 105 |
| Figura 4.14: Resultados simulados do atraso com as políticas activas. ....  | 106 |
| Figura 5.1: Exemplo de operação do protocolo RSVP para estabelecer uma<br>reserva de recursos entre um emissor e um receptor. ....        | 111 |
| Figura 5.2: Estrutura do campo DS dos pacotes IP. ....  | 113 |

---

|   |     |
|---|-----|
| Figura 5.3: Funções de condicionamento de tráfego. ....   | 114 |
| Figura 5.4: Classes DiffServ, Codepoints e PHBs.....  | 115 |
| Figura 5.5: Estabelecimento de uma reserva através do Corretor de Largura de Banda.....   | 116 |
| Figura 5.6: Serviços e acordos.....   | 119 |
| Figura 5.7: Exemplo de fluxo máximo no corte mínimo de uma rede.....  | 123 |
| Figura 5.8: Variação do tempo médio que um pacote demora num sistema de fila de espera em função da intensidade de tráfego $\rho$ .....                 | 124 |
| Figura 5.9: Probabilidade de descarte de pacotes no mecanismo de Detecção Antecipada Aleatória (RED). ....  | 130 |
| Figura 5.10: Variação do tamanho médio da fila de espera.....   | 131 |
| Figura 5.11: Limiares RED para uma marcação com três cores.....   | 133 |
| Figura 5.12: Topologia do núcleo da rede simulada.....  | 137 |
| Figura 5.13: Modelo de fontes de tráfego ligadas a um <i>router</i> do núcleo.....  | 138 |
| Figura 5.14: Rede no <i>Network Animator</i> . ....   | 139 |
| Figura 5.15: Políticas activas utilizadas e as suas relações.....   | 147 |
| Figura 5.16: Funcionamento da regra de ajuste do peso da classe EF.....   | 149 |
| Figura 5.17: Funcionamento da regra de ajuste do peso da classe AF. ....  | 150 |
| Figura 5.18: Comparação do atraso médio com e sem a política activa de gestão dos pesos das classes de tráfego.....                                     | 152 |
| Figura 5.19: Comparação do descarte de pacotes com e sem a política activa de gestão dos pesos das classes de tráfego.....                              | 152 |
| Figura 5.20: Funcionamento da regra de ajuste da largura de banda.....  | 155 |
| Figura 5.21: Evolução do parâmetro $QoS(fila)$ .....  | 158 |
| Figura 5.22: Evolução do parâmetro $QoS(largura\ de\ banda)$ .....  | 158 |
| Figura 5.23: Comparação da largura de banda média e da largura de banda média utilizada com e sem a política activa de gestão de largura de banda. .... | 159 |
| Figura 5.24: Comparação do número de ligações com e sem a política activa de gestão de largura de banda.....  | 160 |
| Figura 5.25: Comparação do débito com e sem a política activa de gestão de largura de banda. ....   | 162 |

---

|   |     |
|---|-----|
| Figura 5.26: Comparação do atraso com e sem a política activa de gestão de largura de banda.....  | 162 |
| Figura 5.27: Comparação do <i>jitter</i> com e sem a política activa de gestão de largura de banda.....   | 163 |
| Figura 5.28: Funcionamento da regra de ajuste do número máximo de ligações em AF. ....  | 166 |
| Figura 5.29: Comparação do número de ligações com e sem a política activa de gestão de admissão de ligações.....  | 167 |
| Figura 5.30: Comparação do débito com e sem a política activa de gestão de admissão de ligações.....  | 168 |
| Figura 5.31: Comparação do atraso com e sem a política activa de gestão de admissão de ligações.....  | 168 |
| Figura 5.32: Comparação do <i>jitter</i> com e sem a política activa de gestão de admissão de ligações.....   | 169 |
| Figura 5.33: Topologia do núcleo da rede com indicação das linhas extra utilizadas. ....  | 172 |
| Figura 5.34: Funcionamento da política de activação de linhas extra.....  | 174 |
| Figura 5.35: Comparação do atraso com e sem a política activa de gestão de linhas extra.....  | 174 |
| Figura 5.36: Funcionamento da regra de ajuste de <i>AFOutPercent</i> .....  | 176 |
| Figura 5.37: Comparação da fracção do número de ligações com a classe de tráfego degradada nas situações com e sem a política activa de gestão do mapeamento para PHBs..... | 179 |
| Figura 5.38: Comparação do número de ligações nas situações com e sem a política activa de gestão do mapeamento para PHBs. ....   | 180 |
| Figura 5.39: Comparação do débito nas situações com e sem a política activa de gestão do mapeamento para PHBs.....  | 180 |
| Figura 5.40: Comparação do atraso nas situações com e sem a política activa de gestão do mapeamento para PHBs.....  | 181 |
| Figura 5.41: Comparação do <i>jitter</i> nas situações com e sem a política activa de gestão do mapeamento para PHBs.....   | 181 |
| Figura 5.42: Comparação do atraso nas situações com e sem as políticas activas de gestão. ....  | 182 |

---

|  |     |
|--|-----|
| Figura 5.43: Comparação da largura de banda média e da largura de banda média utilizada nas situações com e sem as políticas activas de gestão.....    | 183 |
| Figura 5.44: Comparação do número de ligações nas situações com e sem as políticas activas de gestão. ....   | 184 |
| Figura 5.45: Comparação da fracção do número de ligações com a classe de tráfego degradada nas situações com e sem as políticas activas de gestão..... | 184 |
| Figura 5.46: Comparação do débito nas situações com e sem as políticas activas de gestão. ....   | 185 |
| Figura 5.47: Comparação do <i>jitter</i> nas situações com e sem as políticas activas de gestão. ....  | 185 |
| Figura 5.48: Sistema de controlo dos SLAs.....   | 188 |
| Figura 5.49: Comparação do atraso nas situações com e sem a política activa de planeamento.....  | 190 |
| Figura 5.50: Ponto de funcionamento médio para as diferentes classes de tráfego.....   | 192 |
| Figura B.1: Topologia de um sistema de controlo.....   | 231 |





# LISTA DE TABELAS

|  |     |
|--|-----|
| Tabela 1.1: Evolução da gestão ao longo dos anos. ....   | 3   |
| Tabela 2.1: Níveis de QoS relacionados com as perspectivas ODP.....  | 33  |
| Tabela 2.2: Grupos de QoS, características e especializações. ....   | 34  |
| Tabela 4.1: Dimensionamento dos servidores.....  | 95  |
| Tabela 5.1: Exemplo de distribuição da largura de banda pelas classes de<br>tráfego.....   | 128 |
| Tabela 5.2: Configuração das filas de espera de AF. ....   | 141 |
| Tabela 5.3: Configuração das filas de espera de BE. ....   | 141 |
| Tabela 5.4: Distribuição da utilização de protocolos num ISP e no sistema<br>INDEX.....  | 145 |
| Tabela 5.5: Limiares de descarte de pacotes AF fora de perfil. ....  | 151 |
| Tabela 5.6: Dimensionamento do <i>buffer</i> de EF.....  | 157 |
| Tabela 5.7: Dimensionamento do <i>buffer</i> de AF. ....   | 157 |
| Tabela 5.8: Limites utilizados no corretor de largura de banda. ....   | 160 |
| Tabela 5.9: Taxa de sucesso da política activa de gestão de linhas extra na<br>recuperação de falhas. ....                         | 171 |
| Tabela 5.10: Comprimento médio dos percursos no núcleo da rede com<br>diferentes possibilidades de linhas extra ligadas. ....      | 173 |
| Tabela 5.11: Regras de restrição do mapeamento para PHBs.....  | 177 |
| Tabela 5.12: Fracção do tempo que cada regra de restrição do mapeamento para<br>PHBs é usada.....                                  | 178 |
| Tabela 5.13: Resultados obtidos com diferentes valores do SLA para o atraso.....   | 186 |
| Tabela 5.14: Resultados obtidos com diferentes valores do SLA para o atraso e<br>utilização da política activa de planeamento..... | 190 |
| Tabela C.1: Valores de $I^*$ para algumas combinações de $\nu$ e $\alpha$ . ....   | 237 |



# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 O PROBLEMA

Num mundo competitivo como o actual, as empresas precisam de poder oferecer aos seus clientes serviços com uma certa qualidade, sob pena de perderem clientes para empresas concorrentes.

A necessidade de oferecer uma certa qualidade nos serviços torna-se mais premente com o crescimento explosivo da Internet e a utilização generalizada de novos serviços baseados em aplicações sobre esta rede, tais como o comércio electrónico, voz sobre IP, e aplicações multimédia.

A **qualidade de serviço** (QoS) pode ser definida como o efeito colectivo do desempenho de um serviço que determina o grau de satisfação de um utilizador do serviço [E.800].

Como exemplo desta necessidade de qualidade de serviço, apresenta-se um estudo do Gartner Group [HP 99] que indica que o cliente de comércio electrónico tem um conjunto exigente de expectativas, que se podem classificar como de qualidade de serviço. Desde o momento em que o utilizador chega a um sítio na *web*:

- espera uma média de apenas 20 segundos para ver alguma coisa na página;
- espera no máximo 40 segundos para ter a página de abertura completa;
- em 6 segundos decide se fica, ou deixa o sítio;
- utiliza um máximo de 6 cliques com o rato para encontrar a informação desejada;
- o tempo máximo de permanência num sítio é, em média, 12 minutos.

A estas expectativas poderíamos acrescentar algumas de outros tipos, como a disponibilidade do serviço, o custo, e a segurança da transacção.

Levanta-se um problema: Como assegurar requisitos de qualidade de serviço? Alguns dos requisitos dependem da concepção do sistema, outros dependem da forma como o sistema é configurado e opera. O conjunto destas tarefas designa-se habitualmente como **gestão** do sistema.

No resto deste capítulo, apresenta-se o enquadramento deste problema, as propostas desta tese para analisar e assegurar a QoS em tempo real, a estrutura do documento e as notações utilizadas.

## 1.2 ENQUADRAMENTO

A arquitectura TMN (*Telecommunications Management Network*) [M.3010] propõe uma divisão em camadas, chamada de arquitectura de camadas lógicas (*Logical Layered Architecture*, LLA) que se ilustra na figura 1.1. Embora esta arquitectura se refira a gestão de redes de telecomunicações, corresponde a uma hierarquia conceptual que é muito utilizada como termo de comparação pelos autores de gestão de redes.

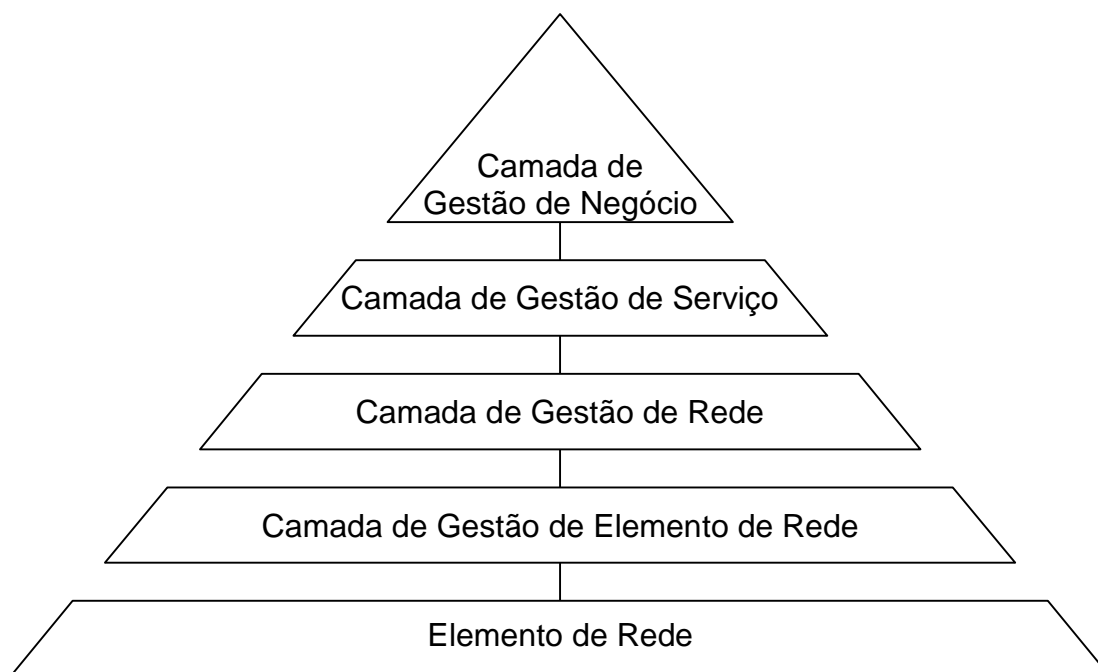


Figura 1.1: Camadas da arquitectura TMN.

A **Camada de Gestão de Elemento de Rede** gere cada elemento de rede numa base individual. Esta camada tem um conjunto de gestores de elemento, cada um responsável por um subconjunto dos elementos de rede.

A **Camada de Gestão de Rede** tem a responsabilidade de gerir todos os elementos de rede, tal como apresentados pela camada de gestão de elemento de rede, tanto individualmente como em conjunto. A camada de gestão de rede não se preocupa com a forma como um elemento particular assegura os serviços internamente. Neste nível, tipicamente tem-se visibilidade sobre toda a rede, sendo necessário manter uma visão independente do vendedor.

A **Camada de Gestão de Serviço** é responsável pelos aspectos contratuais de serviços que estejam a ser fornecidos a clientes, ou que estejam disponíveis a potenciais novos clientes.

A **Camada de Gestão de Negócio** tem responsabilidades sobre toda a empresa, e é o nível ao qual são feitos acordos entre operadores. Esta camada normalmente trata da definição de objectivos, e não de procurar atingir objectivos, mas pode tornar-se fulcro de acção nos casos em que se torne necessária acção executiva.

Estes níveis são complementares, cada um focando um aspecto diferente da gestão da empresa, desde os equipamentos físicos até ao negócio da empresa.

Em termos de implementação da gestão, a evolução tem sido progressiva e rápida. A tabela 1.1 [Lewis 99b] ilustra a evolução da gestão ao longo dos anos.

| Ano    | Objectos Geridos   |
|--------|--|
| 1975   | Equipamentos   |
| 1990   | Redes  |
| 1991   | Tráfego  |
| 1992   | Sistemas   |
| 1994   | Aplicações   |
| 1996   | Empresa (Rede + Tráfego + Sistemas + Aplicações)                     |
| 1998   | Empresa (Rede + Tráfego + Sistemas + Aplicações + Serviço)           |
| futuro | Empresa (Rede + Tráfego + Sistemas + Aplicações + Serviço + Negócio) |

Tabela 1.1: Evolução da gestão ao longo dos anos.

De acordo com a tabela 1.1, nos anos 70, a gestão era feita através de terminais ligados a cada equipamento. Por volta de 1990, apareceram aplicações de gestão que permitiam, a partir de uma consola de gestão centralizada, realizar as tarefas de gestão

da rede. Ainda por volta de 1990, começaram a aparecer produtos que permitiam ver o tráfego que fluía nas linhas da rede. Pouco depois, por volta de 1992, surgiu interesse na gestão de sistemas. Houve esforços no sentido de integrar a gestão de Redes e Sistemas, mas nenhum sistema conseguiu oferecer soluções completas para ambos os casos - situação que se mantém actualmente. Em 1994 surgiu interesse na gestão de aplicações. Claramente, os clientes estavam mais interessados nas suas aplicações do que nas redes, ou nos sistemas que os suportavam. Assim, apareceram sistemas de gestão de aplicações específicas, tais como gestão de SAP/R3 (*Systeme, Anwendungen, Produkte in der datenverarbeitung, Release 3*)<sup>1</sup>, de bases de dados, ou de aplicações Web. Por volta de 1996 surgiram aplicações de gestão de empresa que se limitavam a integrar gestão de redes, de tráfego, de sistemas e de aplicações. Em [Ghetie 97] podem encontrar-se análises detalhadas de várias plataformas de gestão.

No entanto, os executivos não estão preocupados com aspectos técnicos como redes, tráfego, sistemas, ou aplicações, mas sim se os requisitos do seu negócio estão a ser cumpridos. Surge assim, o conceito de **serviço** como uma função providenciada pela rede e sistemas da empresa para realização do negócio da empresa. Exemplos destes serviços são: comércio electrónico e videoconferência. Assim, começam a aparecer produtos para gestão de alguns serviços.

Pensa-se que a evolução futura será no sentido da gestão de processos de negócio. Um **processo de negócio** refere-se à forma como uma empresa coordena e organiza as actividades, a informação e o conhecimento para produzir um bem. Assim, um processo de negócio incluirá serviços, actividades, e relações temporais entre elas. Neste contexto, a gestão incluirá mecanismos para representar, monitorizar e controlar processos de negócio. A figura 1.2 ilustra as relações entre os diferentes tipos de gestão descritos. Em [Lewis 02] chama-se gestão integrada de níveis de serviço à junção dos diferentes níveis de gestão.

---

<sup>1</sup> Terceira versão de soluções colaborativas de negócio electrónico da empresa “Sistemas, Aplicações e Produtos em processamento de dados”.

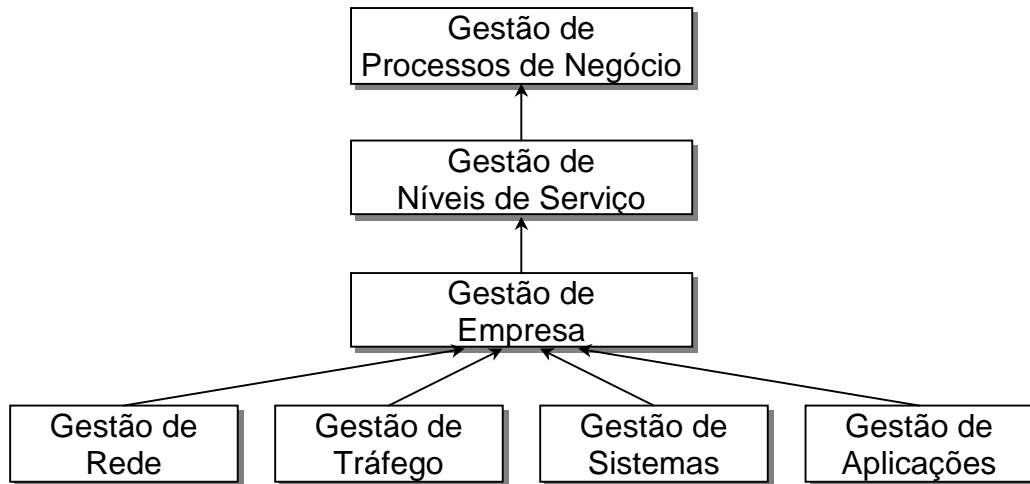


Figura 1.2: Relações entre os diferentes tipos de gestão.

No entanto, os mecanismos que a gestão proporciona para assegurar a qualidade de serviço são ainda rudimentares. Há trabalhos, como se verá em capítulos posteriores, no sentido de proporcionar qualidade de serviço ao nível do tráfego na rede, por exemplo definindo prioridades para o tráfego e reservando recursos. Também é relativamente fácil medir certos parâmetros de qualidade de serviço, por exemplo: tempos de atraso, número de erros, capacidade do sistema, e disponibilidade do serviço. Mas pouco existe sobre como assegurar a qualidade de serviço nos diferentes níveis identificados. Na realidade, em muitos casos a solução passa apenas por uma monitorização e sobredimensionamento dos sistemas envolvidos, faltando a possibilidade de reconfiguração em tempo real.

As **políticas** de gestão são uma forma de atacar os problemas de gestão, que proporciona a pretendida flexibilidade de modificar a estratégia de gestão, ou o comportamento dos sistemas, em tempo real. As políticas podem ser definidas por duas perspectivas que se complementam [RFC 3198]: são objectivos, ou métodos de acção que determinam as decisões presentes e futuras; e ainda um conjunto de regras para administrar, gerir e controlar o acesso aos recursos da rede.

Neste contexto, esta tese propõe uma metodologia para analisar os requisitos de qualidade dos processos de negócio, e uma arquitectura hierárquica de gestão baseada em políticas que actuam sobre os sistemas geridos, cobrindo as várias camadas de gestão da arquitectura TMN, com o objectivo de assegurar esses requisitos. Este modelo

pode ser aplicado nas mais diversas áreas da gestão de redes, apresentando-se exemplos de gestão de redes empresariais e de gestão de redes de dados.

### **1.3 OBJECTIVOS E CONTRIBUIÇÕES**

Os principais objectivos desta tese são:

- Definir e propor uma forma de avaliar a qualidade de serviço oferecida aos utilizadores e de identificar como é que esta qualidade de serviço se reflecte nos diferentes níveis de abstracção.
- Definir e propor uma forma de facilitar a implementação de acordos sobre a qualidade de serviço a ser oferecida aos utilizadores, actuando sobre a rede e os sistemas geridos de maneira a melhor assegurar a qualidade de serviço pretendida.
- Definir e propor uma forma de automatizar as tarefas de gestão, cobrindo monitorização, controlo e planeamento, com vista a fazer cumprir os requisitos de qualidade de serviço pretendidos.

As principais contribuições desta tese são:

- Desenvolvimento de uma arquitectura de gestão com suporte a qualidade de serviço em diferentes níveis de abstracção.
- Uma metodologia para analisar os requisitos de qualidade de serviço e mapeá-los nos diferentes níveis de abstracção.
- Proposta de utilização de políticas activas para gerir a rede e os sistemas com o objectivo de fazer cumprir os requisitos de qualidade de serviço.
- Definição de um conjunto de regras de actuação que podem ser utilizadas como blocos para a construção do comportamento das políticas activas, facilitando-se a sua reutilização.
- Utilização de um mecanismo de ajuste automático dos limiares de funcionamento das políticas activas, com base na história do funcionamento anterior, reconfigurando as políticas activas, sem intervenção humana, de forma a fazer cumprir os requisitos de qualidade de serviço com maior precisão.



## 1.4 ESTRUTURA DA TESE

Com o intuito de alcançar os objectivos propostos, os restantes capítulos e apêndices desta tese estão estruturados da seguinte forma:

O capítulo 2 (“Trabalho Relacionado”) dá uma perspectiva dos conceitos e trabalhos mais importantes relacionados com a área desta tese. São apresentadas as suas virtudes, limitações e a motivação para a tese.

O capítulo 3 (“Modelo de Gestão”) apresenta a proposta desta tese que consiste num modelo de gestão baseado numa hierarquia de políticas activas, com vários níveis de abstracção, para assegurar os requisitos de gestão pretendidos. Este modelo foi inicialmente apresentado em [Pereira 99b].

O capítulo 4 (“Estudo de um Caso - Rede Empresarial”) ilustra a aplicação do modelo de políticas activas ao caso concreto de uma rede empresarial. Começa-se por introduzir o cenário, aplica-se o modelo de gestão ao cenário, explica-se como o cenário foi simulado, como funcionam as políticas activas para gestão da rede e os resultados obtidos. Por fim, avaliam-se os resultados obtidos. Este cenário foi inicialmente apresentado em [Pereira 99b].

O capítulo 5 (“Estudo de um Caso - Fornecedor de Serviço Internet”) ilustra a aplicação do modelo ao caso concreto de um fornecedor de serviço Internet que utiliza tecnologia de serviços diferenciados. Começa-se por fazer uma introdução aos serviços diferenciados, explicando apenas os conceitos necessários para compreender o cenário estudado, pois pretende-se focar a perspectiva de gestão. Depois aplica-se o modelo de gestão ao cenário, explica-se a forma como o cenário foi simulado, o funcionamento das políticas activas para gestão da rede e os resultados obtidos. Finalmente, estuda-se como é possível ajustar automaticamente os limiares de actuação das políticas activas de forma a fazer cumprir com maior exactidão os SLAs, e avaliam-se os resultados obtidos. Partes deste cenário foram inicialmente apresentadas em [Pereira 00a], [Pereira 00b], [Pereira 00c] e [Pereira 01a].

O capítulo 6 (“Avaliação”) avalia o modelo e os cenários propostos, comparando-os com as alternativas existentes na literatura.

O capítulo 7 (“Conclusão”) avalia as contribuições do trabalho desenvolvido, apresenta as conclusões, e aponta sugestões de melhoramentos, extensões e indicações para investigação futura.

O apêndice A (“Lista de Siglas”) apresenta uma lista das siglas mais habituais e os respectivos significados.

O apêndice B (“Introdução aos Controladores PID”) apresenta uma descrição introdutória dos controladores PID. Estes conceitos revelaram-se úteis, pois recorreu-se a um controlador integrativo (isto é, com um integrador) para ajustar dinamicamente os limiares de actuação das políticas activas.

O apêndice C (“Análise Estatística de Resultados de Simulação”) descreve como pode ser avaliado o grau de confiança dos resultados de simulação, e algumas técnicas para melhorar esse grau de confiança.

## 1.5 NOTAÇÕES UTILIZADAS

Para facilitar a interpretação do texto, utilizou-se a seguinte notação:

- Conceitos importantes estão em **negrito**.
- Código de programas, nomes de funções, nomes de objectos, ou outra informação relacionada com a programação está em fonte `Courier`.
- Expressões em línguas estrangeiras estão em *itálico*. Exceptuam-se palavras já adoptadas no português [ACL 01], por exemplo: bit, byte, implementar, interface, monitorizar, software.

# CAPÍTULO 2

## TRABALHO RELACIONADO

Este capítulo dá uma perspectiva dos conceitos e trabalhos mais importantes relacionados com a área desta tese. As primeiras cinco secções apresentam uma introdução à gestão de redes e aos correspondentes modelos funcional, arquitectural, de informação e relacional. Esta primeira parte cobre conceitos base, em muitos casos já normalizados. As restantes secções deste capítulo descrevem conceitos mais específicos, principalmente do domínio da investigação. São analisados os conceitos de qualidade de serviço, gestão de níveis de serviço, políticas de gestão e linguagens de gestão. Ao longo do capítulo são também apresentadas as virtudes e limitações de trabalhos nas diferentes áreas, terminando-se com a motivação para a tese.

### 2.1 INTRODUÇÃO À GESTÃO DE REDES

A gestão de redes envolve a monitorização, análise, controlo e planeamento de actividades e recursos numa rede de modo a oferecer aos utilizadores serviços com um certo nível de qualidade.

Para realizar as tarefas de gestão, torna-se necessário definir [Znaty 94]:

- as funções da aplicação de gestão,
- a arquitectura da rede de gestão,
- a estrutura do sistema de gestão,
- os objectos geridos,
- o método de comunicação a utilizar.

Induzidos por estes objectivos, podem-se definir quatro tipos de modelos que serão descritos com maior detalhe nas secções seguintes.

O **modelo funcional** descreve o conjunto de operações necessárias para atingir um objectivo de gestão. Um exemplo deste modelo é a definição de áreas funcionais de gestão, tal como exposto em [X.700]: gestão de falhas, gestão de configuração, gestão

de contabilização, gestão de desempenho e gestão de segurança. Outro exemplo é a definição de operações mais básicas de gestão, designadas como funções de gestão de sistemas (*System Management Functions*, ou SMF), definidas na série de recomendações X.730-799, e que podem ser utilizadas em diversos serviços de gestão. Exemplos de SMF são funções de reporte de alarmes, funções de gestão de registo de informação (*log*), funções de contabilização de utilização de recursos e funções de sequenciamento de comandos.

O **modelo arquitectural** descreve a estrutura geral das entidades realizando tarefas de gestão, as suas interfaces e os métodos de comunicação. Um exemplo deste modelo é a arquitectura TMN [M.3010], que prevê uma rede de gestão conceptual, interligada com a rede de telecomunicações em diversos pontos para enviar e receber dela informação, bem como controlar a sua operação.

O **modelo de informação** trata da representação lógica dos recursos geridos. Um exemplo deste modelo é a base de dados de informação de gestão (*Management Information Base*, ou MIB) [RFC 1213][X.700][X.701] dos protocolos de gestão da Internet e OSI (*Open Systems Interconnection*), que contém a informação referente aos objectos geridos. Outro exemplo, mais ao nível de meta dados, é a estrutura de informação de gestão (*Structure of Management Information*, ou SMI) [RFC 1155][RFC 2578][X.720], que especifica as regras de definição e construção da MIB, incluindo tipos de dados, representação dos dados e nomes.

O **modelo relacional** descreve o método de comunicação entre o sistema gestor e o sistema gerido, incluindo a pilha de protocolos, a definição dos serviços de gestão e o protocolo de comunicação. Como exemplo deste modelo, tem-se na gestão OSI, o serviço proporcionado pelo CMIS (*Common Management Information Service*) [X.710], que define os procedimentos e parâmetros que as aplicações de gestão podem utilizar em todos os objectos geridos. O CMIS é suportado pelo protocolo CMIP (*Common Management Information Protocol*) [X.711] que realiza a comunicação entre as diferentes entidades.

## 2.2 MODELO FUNCIONAL DE GESTÃO

O modelo funcional de gestão descreve o conjunto de operações necessárias para atingir um objectivo de gestão. Nesta secção são apresentadas as perspectivas operacional e temporal das operações de gestão, as cinco áreas funcionais de gestão e alguns exemplos de funções de gestão de sistemas.

### 2.2.1 Monitorização, Controlo e Planeamento

A figura 2.1 ilustra vários tipos de operações de gestão, cobrindo monitorização, controlo e planeamento.

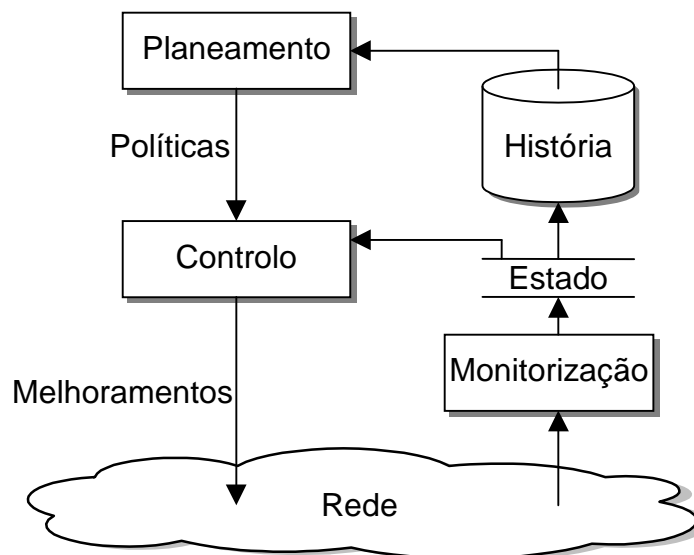


Figura 2.1: Operações de monitorização, controlo e planeamento.

A **Monitorização** corresponde à função mais simples, que é recolher informação sobre o funcionamento da rede. A informação recolhida é mantida numa base de dados que representa o estado actual de funcionamento da rede. Um exemplo de monitorização é a contabilização de custos de utilização de recursos na rede.

O **Controlo** corresponde a uma gestão reactiva, que envolve detectar os problemas, e corrigi-los de forma a manter o serviço em funcionamento. Um exemplo de controlo é a detecção da falha de uma placa e a sua reiniciação.

O **Planeamento** corresponde a uma gestão proactiva, que envolve planear modificações na rede com base na história passada do seu funcionamento e nos

objectivos de gestão definidos em políticas de gestão. A história é obtida guardando numa base de dados vários estados de funcionamento da rede. Um exemplo de planeamento é mudar a localização de um cliente, ou de um servidor, com o objectivo de melhorar a eficiência.

Outra perspectiva para ver as operações de gestão é em termos do horizonte temporal, conforme ilustrado na figura 2.2 [Hegering 99a].

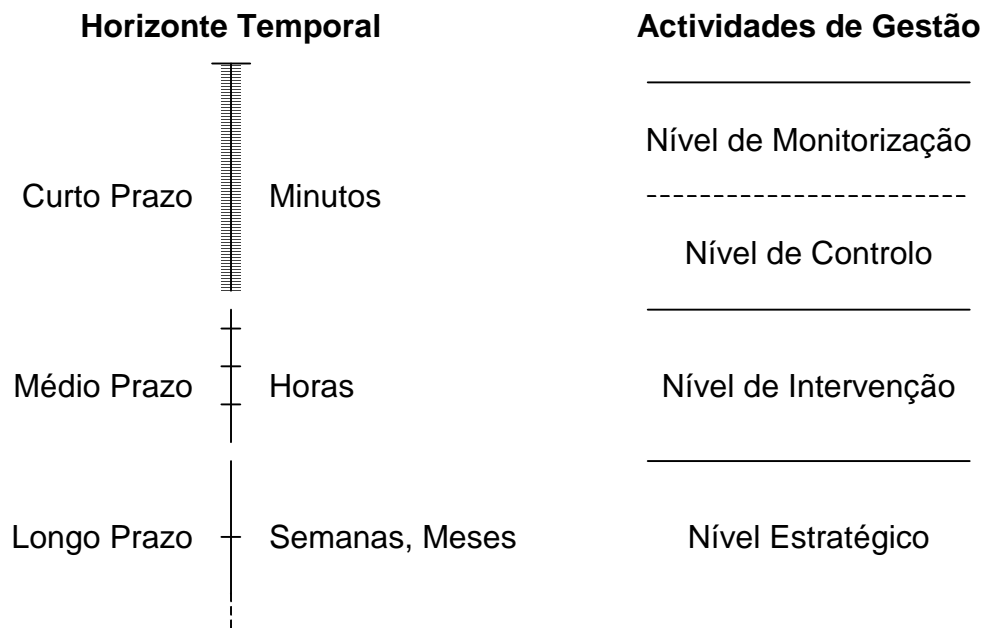


Figura 2.2: Horizontes temporais e actividades de gestão.

As tarefas de **curto prazo** compreendem as tarefas que têm de ser realizadas em segundos ou minutos. Estas incluem tarefas de monitorização que obrigam a determinar num curto intervalo de tempo se objectivos, tais como disponibilidade ou segurança, estão a ser ameaçados. Em termos de actividade de gestão é incluído um nível de controlo para corrigir imediatamente certo tipo de falhas. Um exemplo de uma tarefa de curto prazo é o processamento de falhas para a substituição de um sistema principal por um secundário, que tem de ser realizada num intervalo de tempo muito curto.

As tarefas de **médio prazo** compreendem as tarefas que têm de ser realizadas no período de horas. Enquanto as tarefas de curto prazo têm de estar completamente automatizadas no sistema de gestão para garantir uma resposta curta, as tarefas de médio prazo são habitualmente tarefas de intervenção que são realizadas com a ajuda de um especialista humano. Um exemplo de uma tarefa de médio prazo é o diagnóstico de uma

falha por um especialista com o auxílio de um sistema de diagnóstico ou de gestão de problemas. Outros exemplos são: gerar mudanças de configuração, realizar testes, recolher e avaliar dados de medidas de curto prazo.

As tarefas de **longo prazo** compreendem as tarefas que têm de ser realizadas no período de semanas, ou meses. As tarefas de longo prazo são tarefas estratégicas que utilizam a experiência da operação diária da rede com o objectivo de melhorar o funcionamento para o futuro. Termos importantes neste contexto são análise de tendências e planeamento de capacidade. Um exemplo de tarefa de longo prazo é a manutenção de estatísticas de falhas para seleccionar o melhor fabricante para a compra de novos equipamentos.

As perspectivas operacional e temporal estão relacionadas, na medida em que as funções de monitorização são habitualmente de curto prazo, e as de planeamento de longo prazo. No entanto, enquanto a primeira classificação (em monitorização, controlo e planeamento) corresponde a formas como a gestão actua sobre a rede, a segunda classificação (curto prazo, médio prazo, longo prazo) corresponde à urgência com que a gestão tem de actuar para assegurar os requisitos de qualidade.

Estas duas perspectivas são genéricas, na medida em que se podem aplicar a quaisquer funções de gestão. Na subsecção seguinte concretizam-se as principais categorias de funções que a gestão deve realizar.

### 2.2.2 Áreas Funcionais de Gestão

O modelo OSI define cinco áreas funcionais de gestão [X.700], semelhantes às de TMN [M.3400], que se descrevem de seguida.

A **gestão de falhas** inclui a detecção, isolamento e correcção de funcionamentos anormais. As falhas são condições anormais que conduzem a que os sistemas não cumpram os seus objectivos operacionais, podendo ser persistentes ou transitórias. As falhas manifestam-se como eventos particulares (por exemplo erros) na operação dos sistemas. A gestão de falhas também inclui as funções necessárias para manter e examinar registos de erros, aceitar e actuar perante mensagens de erro, executar testes de diagnóstico, rastrear, isolar e corrigir falhas.

A **gestão de configuração** é o conjunto de funções de gestão que permite identificar, ligar, iniciar, desligar, mudar a configuração, e obter informação sobre mudanças de configuração de objectos geridos.

A **gestão de contabilização** permite identificar custos e atribuir tarifas à utilização de recursos. A gestão de contabilização também inclui funções para informar os utilizadores dos custos ocorridos e dos recursos consumidos, para estabelecer limites de contabilização, e associar planos de tarifas aos recursos.

A **gestão de desempenho** permite avaliar o comportamento dos recursos e a eficiência das actividades de comunicação. A gestão de desempenho também inclui funções para recolher informação estatística, manter e examinar registos do estado do sistema, determinar o desempenho do sistema sob condições naturais e artificiais, e modificar o modo de operação do sistema com o objectivo de realizar tarefas de gestão de desempenho.

A **gestão de segurança** tem como objectivo suportar a aplicação de políticas de segurança através de meios tais como a criação, destruição e controlo de serviços e mecanismos de segurança, a distribuição de informação relevante de segurança, e o relato de eventos relacionados com segurança.

Embora esta divisão tenha sido criada no âmbito da ISO (*International Organization for Standardization*), tem uma grande aceitação por parte de vendedores de sistemas de gestão de redes (tanto normalizados, como não normalizados). Esta divisão é excelente para identificar requisitos funcionais de gestão, mas é difícil que cubra completamente todos os aspectos de uma dada área. Além disso, há alguma sobreposição entre áreas funcionais, como por exemplo na recolha de eventos que existe em todas as áreas. Esta sobreposição leva alguns autores [Raman 98] a propor uma sexta área com as funções comuns. Há outros autores [Caruso 90] que identificam áreas adicionais mais relacionadas com planeamento, tais como: gestão de capacidade, gestão de aprovisionamento, e gestão de administração. Devido a estes factores, a divisão em monitorização, controlo e planeamento apresentada na subsecção anterior (2.2.1) é importante, e complementa estas cinco áreas funcionais.



### 2.2.3 Funções de Gestão de Sistemas

As funções de gestão de sistemas (SMF) oferecem algumas operações genéricas de gestão que funcionam directamente sobre o protocolo de gestão OSI. Estas funções de gestão de sistemas podem ser utilizadas como blocos básicos para construir serviços de gestão comuns a várias aplicações de gestão.

Nesta subsecção apresentam-se, a título de exemplo, três funções de gestão de sistemas que estão mais directamente relacionadas com a área desta tese.

#### 2.2.3.1 SMF de Atributos e Objectos de Medida

A função de gestão de atributos e objectos de medida (*Metric Objects and Attributes*) [X.739] descreve um modelo para monitorização de desempenho e utilização de objectos geridos.

O processo de monitorização executa os seguintes passos:

- **Captura de dados:** periodicamente são extraídos dados dos objectos geridos.
- **Conversão de dados:** os dados extraídos podem ser convertidos, de acordo com um algoritmo, antes de serem utilizados. Por exemplo, pode ser calculada a diferença entre amostras consecutivas.
- **Melhoramento dos dados:** os dados podem ser melhorados através de um algoritmo aritmético para reduzir o efeito de variáveis aleatórias. São sugeridos dois algoritmos: a média móvel de peso exponencial (*Exponential Weighted Moving Average*, ou EWMA), que dá um peso menor aos dados mais antigos; e a média móvel de peso constante (*Uniformly Weighted Moving Average*, ou UWMA), que dá peso igual a todos os dados dentro de uma janela deslizante de dimensão fixa. Depois da operação de melhoramento dos dados, estes ficam visíveis externamente através de atributos de medida.
- **Análise dos dados:** a informação é analisada, e utilizada para cumprir um requisito de desempenho. No caso de monitorização, a análise é uma simples comparação com os limiares para a geração de alarmes utilizando os serviços da função de gestão de reporte de alarmes [X.733].

A figura 2.3 ilustra um exemplo com dois níveis limiares. O limiar 1 é despoletado no sentido ascendente, enquanto o limiar 2 é despoletado quando o valor decresce. É possível controlar se são, ou não, enviados eventos quando o valor medido passa os limiares de activação e de limpeza.

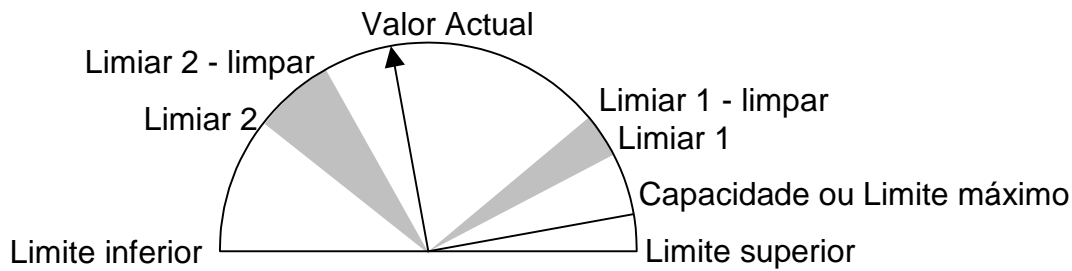


Figura 2.3: Exemplo de limiares da SMF de atributos e objectos de medida.

### 2.2.3.2 SMF de Domínios e Políticas de Gestão

A função de gestão de domínios e políticas de gestão (*Management Domain and Management Policy Management Function*) [X.749] descreve um modelo para a definição de domínios como grupos de objectos geridos, e para a aplicação de políticas de gestão a esses domínios.

Um **domínio** é um objecto de suporte que permite a definição de um conjunto de objectos de interesse para tarefas de gestão. Este agrupamento de objectos geridos pode ser feito por motivos organizacionais ou funcionais. Um exemplo de agrupamento organizacional é a divisão dos objectos geridos em grupos de acordo com áreas geográficas. Um exemplo de agrupamento funcional é a divisão dos objectos geridos em grupos de acordo com a actividade de gestão, ou a área funcional de responsabilidade (falhas, configuração, contabilização, desempenho e segurança).

O conjunto dos membros de um domínio é a união do conjunto dos seus membros directos com o conjunto dos seus membros indirectos. Os membros directos de um domínio são definidos por uma lista de selecção de objectos, eventualmente com um filtro de selecção de membros. Um domínio também permite a referência por nome a outros domínios. Os elementos de domínios assim referidos constituem o conjunto dos membros indirectos do domínio.

Uma **política** é um objecto de suporte que representa um objectivo de gestão, encapsulando a representação de objectivos de gestão de sistemas como regras e semânticas de combinação de regras. As políticas reflectem obrigações, autorizações e aspirações. A sua representação pode ser transparente ou opaca. Uma política

transparente dá acesso à representação da sua semântica, enquanto uma política opaca não dá.

Uma **jurisdição** é um objecto de suporte que representa a relação entre uma política e um domínio, indicando que a política deve ser aplicada aos objectos que fazem parte do domínio.

A figura 2.4 ilustra um exemplo com alguns domínios e políticas associadas. A jurisdição J1 associa a política P1 com o domínio D1. A política P1 aplica-se aos objectos A, B, C, D, E e F, especificados como membros indirectos através do domínio D1. O objecto F está sujeito às políticas P1, P2 e P3.

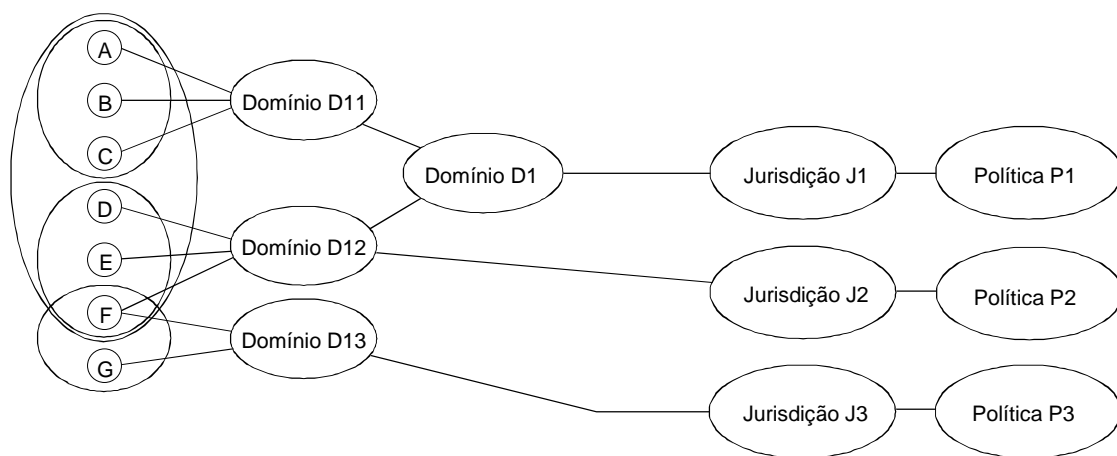


Figura 2.4: Exemplo de domínios e políticas associadas.

Uma **política escalonada** é um objecto que representa, para efeitos de gestão, uma política sujeita a escalonamento. A classe correspondente é derivada da classe política com adição de informação relativa ao escalonador externo.

Uma **política com asserção de valores** é uma política constituída por regras especificadas por asserções de valores. A classe correspondente é derivada da classe política. Uma asserção pode ser de um dos quatro tipos seguintes:

- asserções especificando que operações podem ser realizadas num objecto gerido e os valores permitidos para essas operações;
- asserções especificando valores de atributos de objectos geridos;
- asserções especificando parâmetros em notificações emitidas por um objecto gerido;
- asserções especificando parâmetros em respostas a operações realizadas em objectos geridos.

As asserções são, portanto, expressões lógicas que podem ser avaliadas como verdadeiras, ou falsas. As asserções podem ser combinadas com os operadores lógicos **e**, **ou** e **não**. Sempre que o resultado da avaliação da expressão for falso, há uma violação da política, sendo enviada uma notificação do evento para o gestor.

### **2.2.3.3 SMF de Sequenciamento de Comandos**

O sequenciador de comandos (*Command Sequencer for Systems Management*) [X.753] foi projectado para suportar o escalonamento e execução diferida de operações de gestão de sistemas, a delegação e controlo de actividades de gestão. Esta norma também define a linguagem SMSL (*Systems Management Scripting Language*), baseada na linguagem orientada para objectos C++, para permitir descrever procedimentos para realizar operações de gestão.

### **2.2.3.4 Discussão**

As SMFs definem apenas alguns conceitos básicos de gestão para serem utilizados em aplicações específicas de gestão de redes e sistemas, não dando nenhuma indicação sobre como realizar as tarefas de gestão necessárias.

Por exemplo, a SMF de domínios e políticas de gestão apenas define os conceitos de domínios, políticas, associações entre domínios e políticas (jurisdições) e algumas especializações de políticas. Nesta SMF não é dada nenhuma indicação sobre como organizar os domínios, executar tarefas de gestão com políticas, ou mesmo detectar conflitos entre políticas.

## **2.3 MODELO ARQUITECTURAL DE GESTÃO**

O modelo arquitectural descreve a estrutura geral das entidades realizando tarefas de gestão, as suas interfaces e os métodos de comunicação. Nesta secção começa-se por descrever a arquitectura centralizada, comparando-se, em seguida, com os vários tipos de arquitecturas distribuídas.

### **2.3.1 Arquitecturas Centralizadas**

Dos diferentes aspectos expostos na secção anterior (2.2), referente ao modelo funcional de gestão, identificam-se várias necessidades:

- ter informação sobre o funcionamento de cada entidade da rede;
- utilizar essa informação para agir sobre as entidades da rede;
- interligar as entidades da rede ao gestor para transferir informação sobre o seu funcionamento e transmitir as ordens para as controlar.

Assim identificam-se os seguintes componentes mínimos de uma arquitectura de gestão, ilustrados na figura 2.5:

- **Gestor**, que recolhe a informação dos agentes, e manda realizar as operações necessárias à gestão. É composto de várias partes: interface com o utilizador, aplicação de gestão e interface com o protocolo de transporte.
- **Agente**, que existe em cada equipamento a ser gerido, e mantém a informação necessária à gestão. Realiza as operações de gestão sobre os objectos geridos de acordo com as instruções do gestor, podendo também enviar-lhe notificações de eventos.

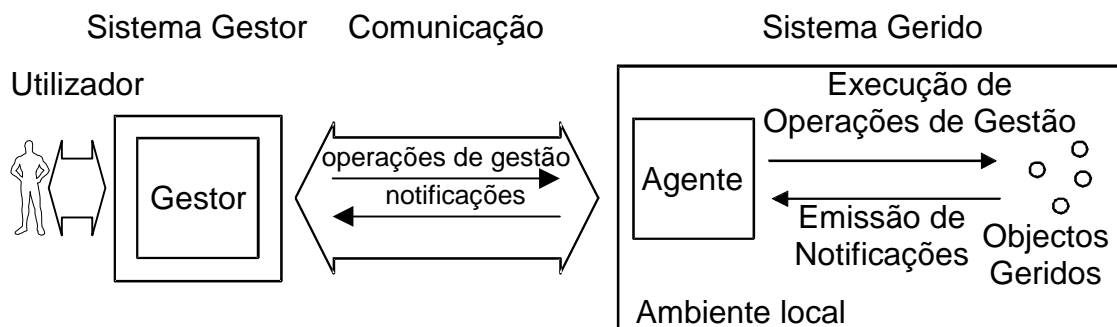


Figura 2.5: Arquitectura gestor / agente.

Esta arquitectura básica, ilustrada na figura 2.5, do tipo cliente-servidor, é comum a vários modelos de gestão, destacando-se a gestão OSI [X.701], TMN [M.3010], a gestão Internet na sua versão 1 [RFC 1157], versão 2 [RFC 1909] e versão 3 [RFC 3411].

Uma arquitectura centralizada, como esta, inclui uma única estação de gestão e uma colecção de agentes locais a cada equipamento gerido. A única estação de gestão tem a aplicação de gestão que concentra todas as capacidades de processamento. Os agentes locais a cada equipamento não têm qualquer inteligência, limitando-se a permitir recolher dados remotamente. Desta forma, uma arquitectura centralizada tem a vantagem de ser simples, mas proporciona uma baixa flexibilidade e uma reduzida

possibilidade de reconfiguração [Baldi 97]. Além disso, este tipo de solução não é escalável para redes muito grandes, e gera congestão na zona da rede próxima da estação de gestão.

### 2.3.2 Arquitecturas Distribuídas

A descentralização das tarefas de gestão é útil para tarefas que beneficiem da distribuição. É possível definir métricas [Meyer 95], ilustradas na figura 2.6, que permitam avaliar se uma aplicação de gestão deve ser implementada de forma centralizada, ou distribuída.

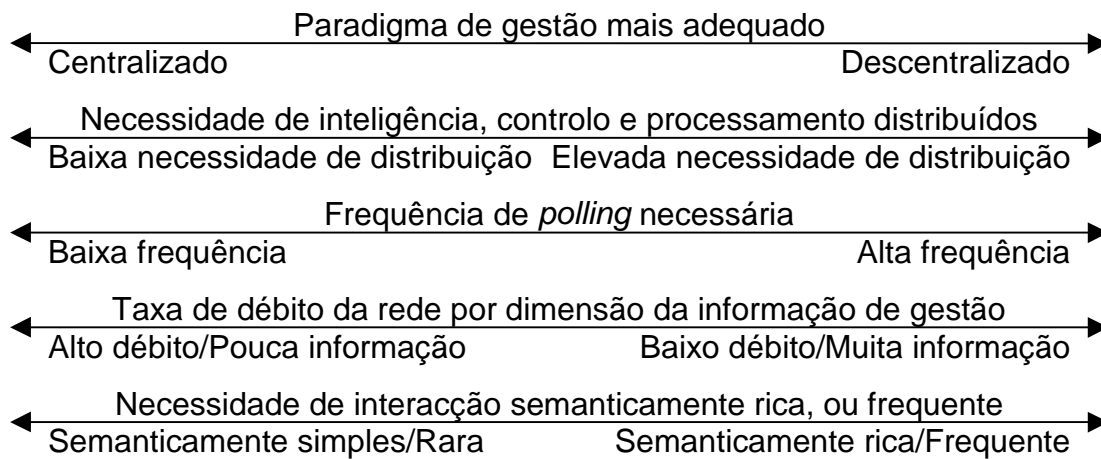


Figura 2.6: Métricas para avaliar benefícios da descentralização.

Certas tarefas de gestão precisam de inteligência, controlo e processamento distribuídos. Exemplos são aplicações que necessitem de respostas rápidas baseadas em informação local, e aplicações que manipulem grandes quantidades de informação para produzir poucos resultados. A distribuição também melhora a escalabilidade e a tolerância a falhas dos sistemas de gestão. No entanto, a distribuição exige uma maior complexidade dos agentes nos equipamentos, o que nos dias de hoje praticamente não constitui problema.

A necessidade de estar perto da informação e a frequência dos *polls* (consultas aos agentes) pode obrigar a que os cálculos sejam realizados localmente aos agentes, correspondendo a uma arquitectura distribuída. Um exemplo é uma função para determinar a saúde da rede que dependa da capacidade de detectar picos de alta frequência nas variáveis.

Se a capacidade disponível na rede face à quantidade de informação a transferir for grande, pode-se recorrer a uma solução centralizada. Caso se tenha uma ligação de baixo débito (por exemplo uma ligação telefónica a um local distante), tem de se optar por uma solução de gestão descentralizada, com processamento junto dos equipamentos.

Finalmente, aplicações que necessitem de interacções frequentes, ou com transferência de estruturas de dados complexas, *scripts*, ou mesmo executáveis, também têm vantagem numa solução distribuída.

A forma mais simples de descentralização é a monitorização remota proporcionada por sondas RMON (*Remote Network Monitoring*) [Stallings 99]. Estas sondas analisam o tráfego num segmento de rede, podendo realizar autonomamente um conjunto predefinido de estudos que podem ser configurados remotamente. Adicionalmente, é possível configurar alarmes na sonda, que geram o envio de uma notificação para o gestor, quando uma situação particular é detectada.

A gestão distribuída é baseada na delegação de tarefas a outras entidades. A gestão por delegação (*Management by Delegation*, ou MbD [Goldszmidt 95]) consiste no envio de programas de gestão para agentes onde executam tarefas de gestão. A SMF de sequenciamento de comandos (descrita na secção 2.2.3.3), e a *script* MIB [RFC 3165] são duas formas de realizar a gestão por delegação.

Para suportar gestão distribuída, as arquitecturas distribuídas onde as aplicações de gestão se sustentam podem ser divididas [Flatin 99] em **arquitecturas hierárquicas** e **arquitecturas cooperantes**. Nas arquitecturas hierárquicas há delegação vertical de tarefas dos níveis hierárquicos superiores nos inferiores. Por outro lado, nas arquitecturas cooperantes há delegação horizontal de tarefas entre entidades pares no mesmo nível hierárquico.

Uma arquitectura hierárquica de gestão é caracterizada por uma árvore de gestores, conforme ilustrado na figura 2.7, onde os agentes se encontram no nível mais baixo. No nível de topo é executada a aplicação de gestão que normalmente delega tarefas nos níveis inferiores, podendo em casos excepcionais actuar directamente sobre os agentes. Os gestores de nível intermédio recebem as aplicações delegadas, e têm responsabilidade de executar tarefas de gestão sobre os níveis inferiores, sob controlo do nível superior. Finalmente, os agentes no nível inferior apenas contêm os objectos geridos sobre os quais são executadas as tarefas de gestão. Em [Siegl 96] é apresentado um exemplo de uma arquitectura deste tipo.

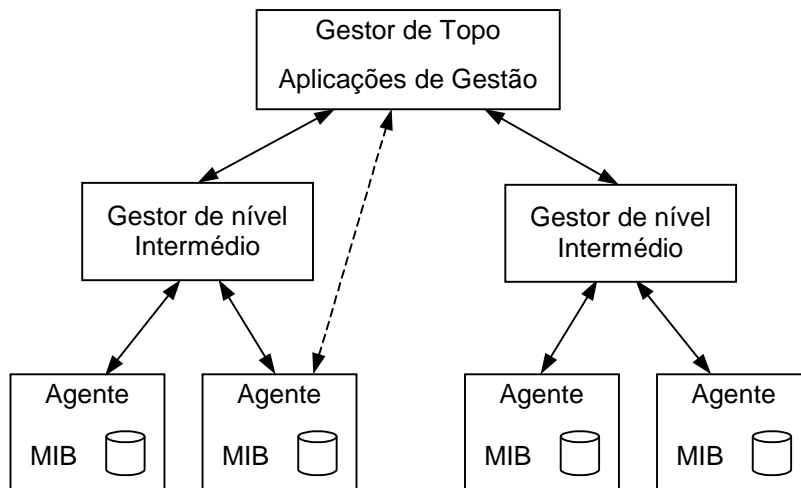


Figura 2.7: Arquitectura hierárquica de gestão.

Nas arquitecturas cooperantes, os agentes normalmente trabalham por objectivos de gestão, tendo autonomia para cooperar com os outros agentes para atingir esses objectivos. Um exemplo de uma arquitectura cooperante está ilustrado na figura 2.8, onde a estação de gestão injecta um programa na rede que percorre vários nós, recolhendo e processando a informação, podendo voltar à estação gestora com o resultado pretendido [Baldi 97]. Este programa chama-se **agente móvel**, embora agente neste caso não tenha o significado habitual de agente de gestão de redes, pois é um termo de inteligência artificial. A cooperação pode não se limitar a um itinerário de uma entidade activa, sendo possível combinar processamento local nos vários nós com comunicação entre eles.

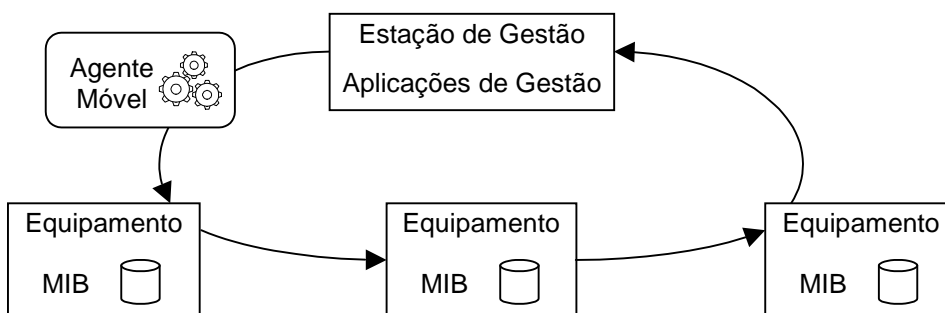


Figura 2.8: Arquitectura de gestão baseada em agentes móveis.



É possível realizar uma combinação das arquitecturas hierárquicas e cooperantes [Schonwalder 96], embora em muitos casos a cooperação se restrinja aos níveis hierárquicos superiores, mantendo-se inalterados os agentes nos equipamentos. Neste trabalho, [Schonwalder 96], é proposta a utilização de SNMP sobre *multicasts* UDP/IP para estabelecer e suportar grupos de agentes cooperantes, sendo definido um algoritmo de eleição para seleccionar o agente mestre que fica com a responsabilidade de supervisionar a execução de *scripts* de gestão delegados. Em [Post 96] também é proposta uma arquitectura mista, definindo o conceito de agência como um grupo de agentes que cooperam para a solução de um problema de gestão. Nesta arquitectura, a comunicação entre o gestor e as agências é realizada através de um protocolo de *multicast* para o grupo de agentes que constituem cada agência. Outro exemplo, com estrutura diferente, é dado em [Sahai 97] que combina uma arquitectura hierárquica com uma cooperante baseada em agentes móveis.

As arquitecturas distribuídas também podem ser divididas [Flatin 99] em: arquitecturas fracamente distribuídas e arquitecturas fortemente distribuídas. Nas arquitecturas fracamente distribuídas, o processamento de gestão está concentrado num conjunto restrito de estações de gestão. A gestão utilizando sondas RMON é um exemplo de uma arquitectura fracamente distribuída. Por outro lado, nas arquitecturas fortemente distribuídas, o processamento está repartido por cada agente. A gestão utilizando agentes móveis é um exemplo de uma arquitectura fortemente distribuída.

As **redes activas** [Tennenhouse 97] são outro exemplo de arquitectura fortemente distribuída. A ideia principal das redes activas é que os pacotes que as percorrem levam não só dados, como também programas. Há duas aproximações à realização das redes activas. A aproximação dos comutadores programáveis mantém o formato dos pacotes existente, proporcionando um mecanismo para carregar programas nos nós da rede. Por outro lado, a aproximação das cápsulas vai mais longe, substituindo os pacotes passivos das redes de hoje por programas activos miniatura que são encapsulados nas tramas e executados em cada nó ao longo do seu caminho. A aproximação dos comutadores programáveis é semelhante à delegação de programas da gestão por delegação. A aproximação das cápsulas, embora semelhante aos agentes móveis, é mais geral, pois prevê a existência de processamento no nível rede, e não apenas como uma aplicação [Psounis 99].

Pensa-se que no futuro as arquitecturas de gestão de redes e sistemas serão cada vez mais distribuídas e flexíveis. Assim, uma arquitectura de gestão deve poder adaptar-se dinamicamente, tanto a diferentes topologias de rede, como a diferentes necessidades de processamento de tarefas de gestão [Pereira 98].

## 2.4 MODELO DE INFORMAÇÃO DE GESTÃO

O modelo de informação trata da representação lógica dos recursos geridos. O modelo deve descrever a forma de aceder à informação de gestão, o significado da informação e o modo de definir nova informação de gestão.

A base de dados de informação de gestão (*Management Information Base*, ou MIB) representa o repositório com a informação de gestão que existe num agente. Os vários objectos geridos que constituem a MIB estão organizados numa árvore, sendo atribuído um identificador único a cada um. Este espaço de nomes é global e muito rígido, sendo a definição feita em normas. A figura 2.9 ilustra a representação de parte da tabela de encaminhamento na MIB-II [RFC 1213], a segunda versão da MIB na gestão Internet. O objecto `ipInReceives`, por exemplo, corresponde ao identificador de objecto 1.3.6.1.2.1.4.3, de acordo com o caminho na árvore até lhe chegar<sup>1</sup>. Também se pode usar os nomes correspondentes: `iso.org.dod.internet.mgmt.mib-2.ip.inReceives`.

A estrutura de informação de gestão (*Structure of Management Information*, ou SMI) especifica as regras de definição e construção da MIB, incluindo tipos de dados, representação dos dados, relações entre objectos geridos, e nomes. A descrição dos objectos geridos de acordo com a SMI Internet [RFC 1155][RFC 2578] é feita utilizando um subconjunto da sintaxe de ASN.1 (*Abstract Syntax Notation One*) [X.208]. A figura 2.10 ilustra a descrição do objecto `ipInReceives`, onde é definido:

- o tipo do objecto, como sendo um contador, que representa um inteiro não negativo;
- o modo de acesso ao objecto, como sendo apenas de leitura;
- o suporte de realização, como devendo ser obrigatoriamente suportado;
- a descrição textual da semântica do objecto;

---

<sup>1</sup> Na realidade, este identificador corresponde ao tipo de objecto, sendo a instância do objecto obtida concatenando “.0” no fim. Neste caso, obter-se-ia 1.3.6.1.2.1.4.3.0 para a instância do objecto.

- a posição respectiva na MIB, como o terceiro objecto debaixo do nó ip.

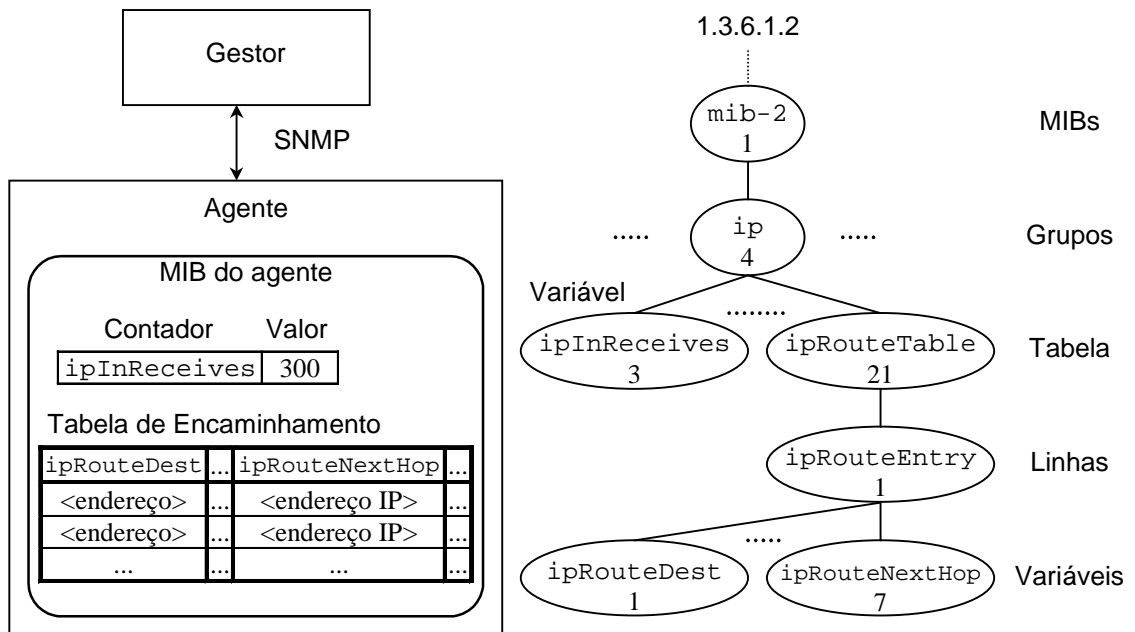


Figura 2.9: Exemplo de MIB Internet.

```

ipInReceives OBJECT-TYPE
    SYNTAX Counter
    ACCESS read-only
    STATUS mandatory
    DESCRIPTION
        "The total number of input datagrams received from
        interfaces, including those received in error."
    ::= { ip 3 }
    
```

Figura 2.10: Exemplo de descrição de objecto gerido na SMI Internet.

Existem outros modelos de informação de gestão semelhantes ao da Internet, mas normalizados por outras entidades, destacando-se o CIM (*Common Information Model*) da DMTF (*Distributed Management Task Force*), e o SMI OSI.

A especificação do CIM [CIM 99] define:

- um meta esquema com a definição formal do modelo, os termos utilizados, a forma de utilização e semântica;

- uma linguagem de descrição da informação de gestão, a MOF (*Managed Object Format*);
- um modelo de informação, constituído por conjuntos de classes com a informação de gestão.

O SMI OSI é composto por três partes principais:

- o modelo de informação de gestão [X.720], que define os conceitos fundamentais de objectos geridos;
- a definição de informação de gestão [X.721], que contém as definições das principais classes de objectos geridos;
- as directivas para definição de objectos geridos [X.722], que dão assistência prática à criação de novas classes com informação de gestão.

A normalização da informação de gestão é especialmente importante para permitir que quaisquer aplicações de gestão possam utilizar a informação de gestão recolhida dos objectos geridos de qualquer fabricante.

## **2.5 MODELO RELACIONAL DE GESTÃO**

O modelo relacional descreve o método de comunicação entre o sistema gestor e o sistema gerido. Nesta secção apresentam-se os protocolos OSI e Internet, os mais utilizados. De seguida, refere-se a utilização de tecnologias de objectos distribuídos para a comunicação. Referem-se ainda outros protocolos de gestão e formas de garantir a segurança da comunicação. Por fim, faz-se uma análise crítica dos diferentes sistemas.

### **2.5.1 Protocolos de Gestão OSI e Internet**

A figura 2.11 ilustra as pilhas de protocolos utilizadas na gestão OSI e Internet, verificando-se que na gestão Internet a pilha é muito mais simples.

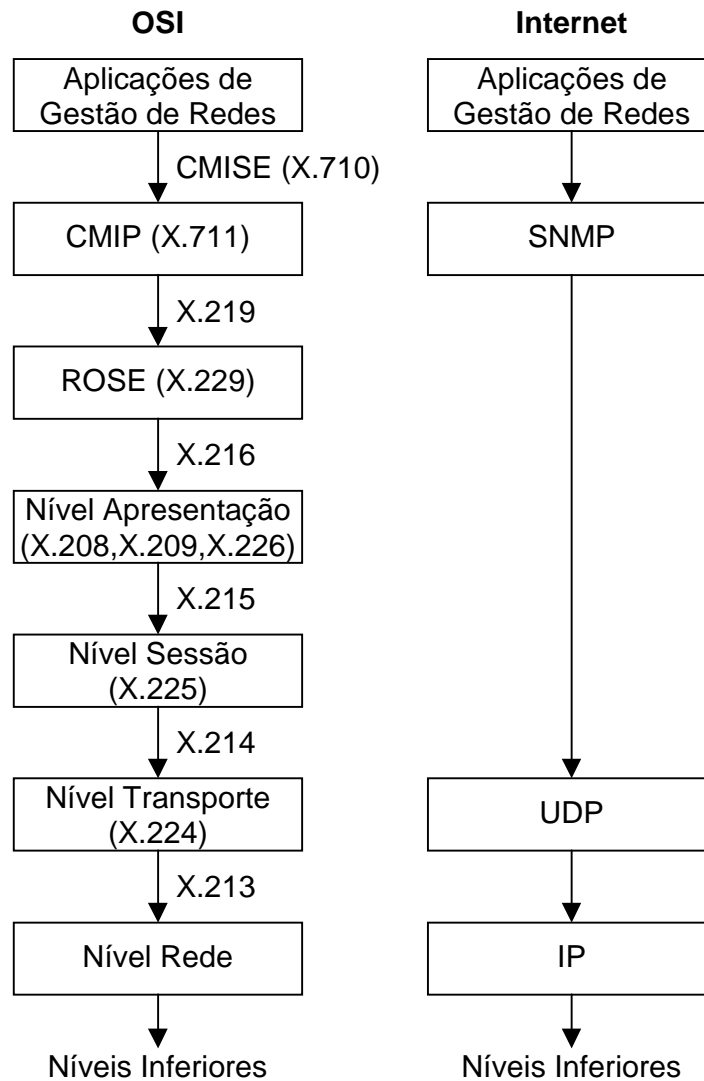


Figura 2.11: Pilhas de protocolos na gestão OSI e Internet.

O funcionamento do protocolo de gestão está ilustrado na figura 2.12 para o caso de uma operação GET no protocolo OSI. A operação GET permite obter informação sobre objectos geridos. O funcionamento para o protocolo SNMP e para a operação SET, que permite modificar informação nos objectos geridos (tanto no caso OSI, como no caso Internet), é semelhante ao ilustrado. As PDUs (*Protocol Data Unit*) do protocolo são habitualmente codificadas de acordo com as regras de codificação básicas de ASN.1 [X.209].

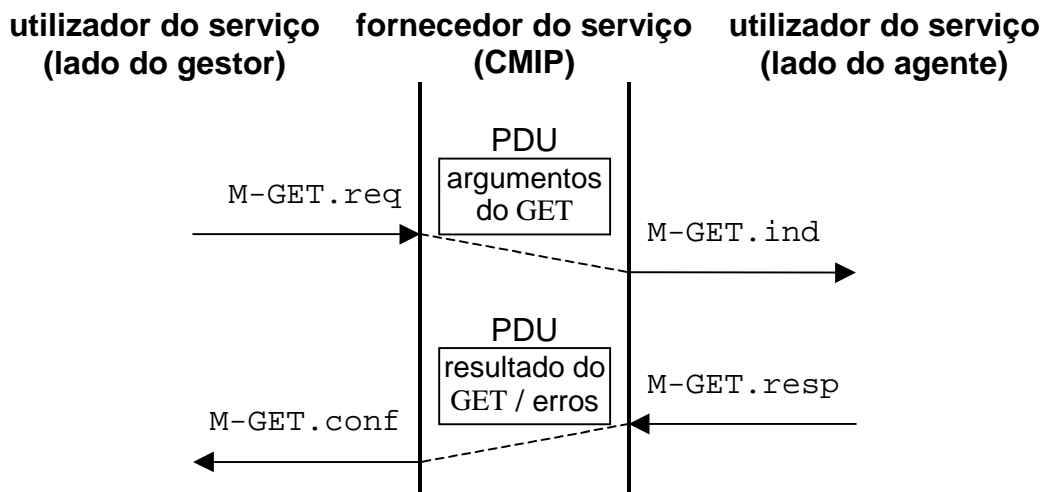


Figura 2.12: Operação do protocolo de gestão OSI para a primitiva GET.

As operações GET e SET são as mais simples, sendo comuns a vários protocolos, havendo outras operações específicas a protocolos particulares. Exemplos de outras operações são: o envio de notificações por parte dos objectos geridos (reporte de eventos no CMIP, e PDUs `trap` no SNMP), quando é detectada alguma situação relevante para o gestor, permitindo reduzir a frequência das monitorizações necessárias e, por consequência, o tráfego de gestão; a comunicação entre gestores através de PDUs `InformRequest` no SNMPv2 [RFC 1905], e versões posteriores; a invocação de operações sobre objectos geridos no CMIP, sendo habitual emular estas invocações em SNMP usando o SET com valores com significados especiais [RFC 1212]; a criação e destruição de instâncias de objectos geridos no CMIP, que também é habitualmente emulada em SNMP usando o SET.

## 2.5.2 Gestão usando Objectos Distribuídos

Uma forma alternativa de realizar a comunicação entre o sistema gestor e os objectos geridos é através da invocação remota de métodos de objectos.

A arquitectura CORBA (*Common Object Request Broker Architecture*) [CORBA 01] é um exemplo de uma arquitectura que se baseia numa tecnologia de objectos distribuídos. Esta arquitectura oferece um conjunto de ferramentas e serviços

para suportar a comunicação entre objectos distribuídos de forma transparente, sendo, por isso, mais do que um simples protocolo de comunicação<sup>2</sup>.

O ambiente de processamento distribuído proporcionado pela arquitectura CORBA também pode ser utilizado para aplicações de gestão de redes e sistemas. O *Tivoli Management Environment* [Orfali 96a][Tivoli] é um exemplo de uma arquitectura comercial de gestão de sistemas baseada na arquitectura CORBA. A TINA (*Telecommunications Information Networking Architecture*) [Pavón 98] é um exemplo de uma arquitectura para o desenvolvimento de serviços e aplicações de gestão de redes, utilizada pelos operadores de telecomunicações e fabricantes, que também usa a arquitectura CORBA.

A interacção entre CORBA e CMIP ou SNMP foi tratada pelo grupo JIDM (*Joint Inter-Domain Management*) [JIDM 00] para permitir trabalhar com os sistemas já existentes. Para permitir a interacção transparente, sem que os objectos envolvidos tenham conhecimento da possível tradução, é necessário definir dois componentes:

- um esquema de tradução entre os diferentes modelos de objectos de gestão, chamado de **Tradução de Especificação**;
- um mecanismo de tradução entre os protocolos e comportamentos utilizados em ambos os domínios de gestão, chamado de **Tradução de Interacção** [CORBA TMN 00].

### 2.5.3 Outros Protocolos de Gestão

Existem outros sistemas de gestão de redes baseados em outros protocolos. Um exemplo digno de relevo é a iniciativa WBEM (*Web-Based Enterprise Management*) [WBEM], promovida por numerosas empresas, onde se inclui a Microsoft. A WBEM propõe: um modelo de informação de gestão, o CIM, já referido na secção 2.4; um modelo de codificação, o xmlCIM [XMLCIM 99], que define uma gramática XML (*Extensible Markup Language*) [XML 00] para representar mensagens e declarações CIM; e um mecanismo de transporte, definindo o mapeamento de operações CIM sobre HTTP (*Hypertext Transfer Protocol*) [CIMHTTP 99].

---

<sup>2</sup> Na realidade, a comunicação entre objectos baseia-se numa interface de programação entre objectos clientes e objectos servidores, e não tanto num protocolo de comunicação.

#### 2.5.4 Segurança

Sendo as operações de gestão críticas para o correcto funcionamento de uma rede e dos sistemas nela envolvidos, torna-se necessário protegê-las contra ataques ou uso indevido. As protecções de segurança devem estar incluídas no protocolo de gestão de redes, de forma a serem transparentes para as aplicações de gestão. No entanto, como os mecanismos de segurança reduzem a eficiência dos sistemas envolvidos, em muitos casos não são utilizados.

Os métodos habitualmente utilizados para garantir a segurança das operações de gestão de redes são a utilização de métodos de autenticação de mensagens e métodos de cifra de mensagens.

Os métodos de autenticação permitem garantir a integridade dos dados e a autenticidade da origem dos dados. São normalmente baseados em assinaturas, certificação, ou notários. Recorrem a algoritmos de síntese (*hashing*), tais como o MD5 (*Message Digest 5*) [RFC 1321], ou o SHA (*Secure Hash Algorithm*) [FIPS 180-1] [RFC 3174].

Os métodos de cifra permitem garantir a confidencialidade dos dados. Exemplos de métodos de cifra são: o DES (*Data Encryption Standard*) [FIPS 46-3] e o RSA (iniciais dos autores Rivest, Shamir e Adleman) [RSA 78]. Note-se que alguns métodos, como é o caso do RSA, podem ser utilizados tanto para cifrar como para autenticar mensagens.

#### 2.5.5 Discussão

Nesta tese procurou-se trabalhar sempre por cima dos protocolos de gestão, pelo que, à partida, qualquer protocolo de gestão poderá ser utilizado. As diferenças principais serão em termos da eficiência e segurança de cada um. Por exemplo: o CMIP é mais poderoso do que o SNMP na geração de notificações, permitindo economizar tráfego de gestão; por outro lado, o CMIP é orientado à ligação, tornando mais complexas operações muito simples; o SNMPv1, pela sua simplicidade, não tem segurança para além de um nome de comunidade que passa em claro nas PDUs, sendo habitual nesta versão do SNMP realizar-se controlo de acessos pelo endereço IP de origem; apesar disso, a simplicidade do SNMP tem uma vantagem – exige relativamente



poucos recursos do equipamento, pelo que pode ser suportado até pelos equipamentos mais simples.

No entanto, a utilização de uma perspectiva de objectos distribuídos como a proporcionada pela arquitectura CORBA traz algumas vantagens [Haggerty 98] [Orfali 96a]. O facto de se utilizar esta arquitectura, que é orientada para objectos, para o desenvolvimento de aplicações torna as aplicações facilmente extensíveis, e com uma interface normalizada independente da linguagem de programação e máquina utilizada. Além disso, proporciona uma solução escalável para grandes redes, e que pode ser integrada com os protocolos de gestão existentes. As maiores exigências de recursos computacionais necessários para suportar esta arquitectura já não constituem problema, visto que o poder de cálculo, mesmo dos equipamentos mais simples, tem aumentado bastante.

Verifica-se que os operadores de telecomunicações estão a migrar para plataformas baseadas em CORBA, como é o caso da TINA [Pavón 98], tanto para desenvolver aplicações de gestão de redes, como para a evolução das redes inteligentes [Mampaey 00].

Assim, será de esperar que, no futuro, o SNMP se mantenha em equipamentos de pequeno porte num ambiente Internet, devido à sua simplicidade, havendo uma evolução das plataformas de gestão para arquitecturas de objectos distribuídos.

Termina-se aqui a descrição dos modelos de gestão de redes, sendo o resto deste capítulo dedicado à apresentação de conceitos mais específicos, necessários para esta tese.

## 2.6 QUALIDADE DE SERVIÇO

A qualidade de serviço (QoS) é definida em [E.800] como “o efeito colectivo do desempenho de um serviço que determina o grau de satisfação de um utilizador do serviço”<sup>3</sup>. Uma outra definição, no contexto de ambientes distribuídos baseados em objectos, é dada em [X.641][X.902] como “um conjunto de qualidades relacionadas com o comportamento colectivo de um ou mais objectos”.

---

<sup>3</sup> A visão da QoS por parte do operador da rede é chamada **qualidade de funcionamento da rede** (*network performance*, ou NP), definida como “a capacidade de a rede fornecer as funções relacionadas com a comunicação entre utilizadores”. Optou-se por não distinguir as duas visões.

A especificação da QoS [Aurrecoechea 98][Campbell 96] pretende representar os requisitos de QoS das aplicações e as políticas de gestão, sendo normalmente diferente consoante a camada do sistema a que diz respeito. Para assegurar os níveis de QoS desejados pelas aplicações, de acordo com a especificação da QoS, são utilizados mecanismos de QoS.

Os mecanismos de QoS podem ser divididos em mecanismos de provisão de QoS e mecanismos de controlo e gestão de QoS. Os mecanismos de provisão de QoS tratam das fases de estabelecimento de fluxos e renegociação de QoS. Os mecanismos de controlo e gestão de QoS tratam da fase de transferência de dados.

Os requisitos de QoS diferem consoante o grau de abstracção em causa, sendo mais qualitativos e gerais ao nível do utilizador, por um lado; e mais quantitativos e técnicos ao nível do sistema, por outro lado. Por vezes é difícil encontrar o grau de abstracção correcto e fazer as correspondências de uns para outros.

Os níveis de QoS podem ser classificados de acordo com as perspectivas do modelo de referência para processamento distribuído aberto (*Open Distributed Processing*, ODP), tal como exemplificado na tabela 2.1 [Leydekkers 96]. Esta classificação indica, para cada perspectiva ODP: o nível de QoS; a forma como a QoS é apercebida pelos utilizadores; e exemplos de parâmetros que descrevem como a QoS é realizada nos equipamentos que suportam os sistemas.

O ODP é uma norma para a definição de sistemas distribuídos em geral. O ODP recomenda a utilização de cinco perspectivas [X.901][X.903] para modelar sistemas distribuídos complexos. Cada perspectiva foca um aspecto particular das características do sistema. As cinco perspectivas são as seguintes:

- **perspectiva de empresa:** foca os objectivos, âmbito e requisitos do sistema;
- **perspectiva de informação:** foca a informação, e o seu processamento, no sistema;
- **perspectiva de computação:** foca a decomposição do sistema num conjunto de objectos que interagem entre si em interfaces, permitindo a distribuição do sistema;
- **perspectiva de engenharia:** foca a infra-estrutura necessária para suportar a distribuição;
- **perspectiva de tecnologia:** foca a escolha da tecnologia (hardware e software) necessária para suportar o sistema.

| Perspectiva do ODP | Nível de QoS        | Características   | Exemplos   |
|--------------------|---------------------|---|--|
| Empresa            | QoS Subjectiva      | Requisitos de QoS do utilizador (informais e subjectivos)   | “O áudio deve ser apresentado em conjunto com o vídeo”;<br>“A qualidade do vídeo deve ser semelhante à da televisão” |
| Informação         | QoS Objectiva       | Definição precisa dos requisitos de QoS derivada da especificação subjectiva. Normalmente independente da aplicação.      | 25 fps (PAL), 30 fps (NTSC),<br>Relação H:V de 4:3 (formato PAL).<br>Qualidade Telefónica: 8 KHz                     |
| Computação         | QoS de Aplicação    | Descreve os requisitos de QoS para as aplicações em termos de qualidade dos <i>media</i> e relações entre os <i>media</i> | Sincronização dos lábios entre o áudio e o vídeo: $\pm 80$ ms.   |
| Engenharia         | QoS do Sistema      | Descreve os requisitos de QoS para o sistema operativo  | Tamanho dos <i>buffers</i> , Operações por segundo, Memória (Mb)   |
|                    | QoS da Rede         | Descreve os requisitos de QoS para a rede   | Débito (Mbit/s), Atraso (ms), Jitter (ms)  |
| Tecnologia         | Propriedades de QoS | Descreve as características de QoS dos dispositivos, sistemas operativos, e tecnologias de rede. Propriedades de QoS.     | dispositivo de vídeo (formato PAL), dispositivo de áudio ( $\mu$ -law), taxa de descarte de células de ATM.          |

Tabela 2.1: Níveis de QoS relacionados com as perspectivas ODP.

Para descrever a QoS, é possível definir vários grupos de características de QoS [X.641], correspondendo a diferentes aspectos quantificáveis de QoS. Estas características de QoS podem, por sua vez, ser especializadas em outras características, tal como descrito na tabela 2.2. Os requisitos de QoS são expressos em parâmetros de QoS, que concretizam valores para as características de QoS. Os parâmetros de QoS podem ser trocados entre diferentes entidades. A tabela 2.2 também indica uma forma possível de fazer a quantificação para as várias características de QoS.

A semântica dos requisitos de QoS pode ser complexa, incluindo [X.641]:

- valores de diferentes parâmetros;
- papel que o valor toma, podendo ser (figura 2.13):
  - um limite superior, ou inferior, que não deve ser ultrapassado;
  - um limiar superior, ou inferior;
  - um objectivo de funcionamento;
  - um parâmetro auxiliar;
- acções a ser desencadeadas quando se atinge um limite, ou limiar;
- a natureza do acordo entre as partes.

| Grupo de Características                 | Nome da Característica      | Nome da Especialização                     | Quantificação    |
|--|-----------------------------|--|------------------|
| Características temporais                | Data/tempo                  |  | Tempo            |
|  | Atraso                      |  | Tempo            |
|  | Duração                     | Duração Remanescente                       |                  |
| Frescura                                 |                             |  | Tempo            |
| Características de coerência             | Coerência temporal          | Coerência temporal de produção de dados    | Booleano         |
|  |                             | Coerência temporal de transmissão de dados | Booleano         |
|  |                             | Coerência temporal de consumo de dados     | Booleano         |
|  | Consistência espacial       | Consistência espacial intemporal           | Booleano         |
|  |                             | Consistência espacial temporal             | Booleano         |
| Características de capacidade            | Capacidade                  | Débito                                     | Taxa, e.g. bit/s |
|  | Débito                      | Taxa de introdução de dados do utilizador  | Taxa, e.g. bit/s |
|  |                             | Débito de informação do utilizador         | Taxa, e.g. bit/s |
|  |                             | Débito de informação da aplicação          | Taxa, e.g. bit/s |
|  |                             | Débito de subsistema                       | Taxa, e.g. bit/s |
|  |                             | Capacidade de processamento                | Instruções/s     |
|  | Capacidade de processamento | Débito do sistema                          | Instruções/s     |
|  | Carga de operação           | Carga de operação                          | Rácio [0..1]     |
|  |                             | Carga de associações                       | Rácio [0..1]     |
|  |                             | Carga de subsistema                        | Rácio [0..1]     |
| Características de integridade           | Precisão                    | Erro de endereçamento                      | Probabilidade    |
|  |                             | Erro de entrega                            | Probabilidade    |
|  |                             | Erro de transferência                      | Probabilidade    |
|  |                             | Erros toleráveis                           | Probabilidade    |
|  |                             | Resiliência                                | Probabilidade    |
|  |                             | Integridade de transferência               | Probabilidade    |
|  |                             | Erro de estabelecimento                    | Probabilidade    |
|  |                             | Erro de recuperação                        | Probabilidade    |
|  |                             | Erro de libertação                         | Probabilidade    |
| Características de segurança             | Segurança                   |  | Nível            |
| Características de protecção (Segurança) | Protecção                   |  | Probabilidade    |
|  | Controlo de acessos         |  | Nível            |
|  | Protecção de dados          |  | Nível            |
|  | Confidencialidade           |  | Nível            |
|  | Autenticidade               |  | Nível            |
| Características de fiabilidade           | Disponibilidade             | Disponibilidade do canal                   | Rácio [0..1]     |
|  |                             | Disponibilidade de ligação                 | Rácio [0..1]     |
|  |                             | Disponibilidade de processamento           | Rácio [0..1]     |
|  | Fiabilidade                 |  | Tempo            |
|  | Contenção de faltas         |  |                  |
|  | Tolerância a faltas         |  |                  |
|  | Reparabilidade              |  | Tempo            |
| Outras características                   | Precedência                 |  | Prioridade       |

Tabela 2.2: Grupos de QoS, características e especializações.

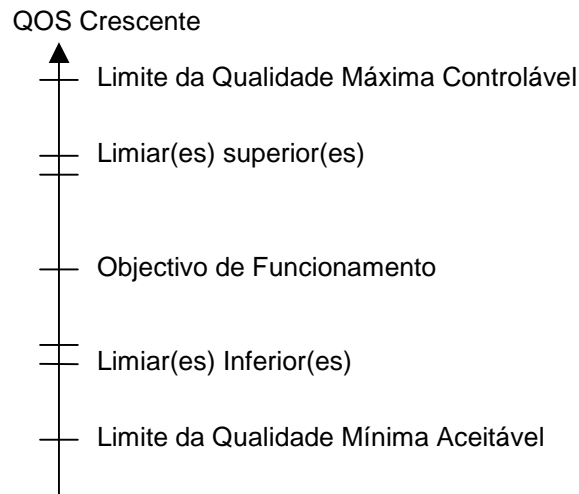


Figura 2.13: Limites e limiares de QoS.

Os limites e limiares superiores podem desencadear acções quando são ultrapassados no sentido crescente, enquanto os limites e limiares inferiores podem desencadear acções quando são ultrapassados no sentido decrescente. Estas acções podem incluir:

- modificar o funcionamento do fornecedor do serviço, ou do utilizador do serviço, de modo a tentar respeitar os limites;
- registar o valor;
- enviar um sinal, por exemplo “aviso” ou “reiniciação”, para o fornecedor do serviço, ou para o utilizador do serviço;
- abortar o serviço.

O fornecedor do serviço e o utilizador do serviço podem acordar tomar acções para manter a QoS. Neste caso, essas acções podem incluir:

- monitorizar a QoS obtida;
- controlar os fluxos de informação;
- actuar para restringir a QoS obtida;
- reservar, ou reafectar recursos;
- avisar que limiares ou limites foram ultrapassados;
- suspender, ou abortar a actividade;
- suspender, ou abortar outras actividades que competem para obter os recursos.

Dada a variedade de acções possíveis, podem ser definidos vários níveis de acordos que especificam as acções que o fornecedor do serviço e o utilizador do serviço devem tomar. Em [X.641] definem-se três níveis de acordos:

- **Melhor esforço**, sem nenhuma garantia que a QoS negociada seja assegurada, não sendo tomada nenhuma medida para monitorizar ou assegurar a QoS.
- **Compulsivo**, onde o serviço é abortado se a QoS se reduzir abaixo do nível negociado, o que pode acontecer se um serviço de maior prioridade necessitar dos recursos.
- **Garantido**, onde a QoS negociada tem de ser garantida, não sendo iniciado o serviço se tal não for possível.

Embora os conceitos de qualidade de serviço sejam gerais, podendo ser aplicados a muitos domínios, verifica-se que os mecanismos de controlo e gestão da QoS existentes são limitados, sendo normalmente aplicados apenas a serviços de nível rede e frequentemente em contextos de aplicações multimédia. Nesta tese procurou-se generalizar os conceitos de QoS para outros níveis de abstracção desde os processos de negócio até aos equipamentos. A gestão de níveis de serviço, que se apresenta na secção seguinte, é o primeiro passo no sentido de aplicar conceitos de QoS a outros níveis de abstracção.

## 2.7 GESTÃO DE NÍVEIS DE SERVIÇO (SLM)

Os trabalhos mais antigos sobre gestão de níveis de serviço (*Service Level Management*, ou SLM) remontam à segunda metade dos anos 80, tendo sido publicados numerosos artigos nas actas das conferências do Computer Measurement Group (CMG) [CMG] desde 1985 [McBride]. Alguns dos conceitos utilizados são semelhantes aos da área de engenharia de desempenho de software (*Software Performance Engineering*, ou SPE), cuja popularidade cresceu em paralelo.

A gestão de níveis de serviço corresponde ao processo iterativo de [Lewis 99a]:  
1) identificar processos de negócio; 2) identificar os serviços de rede dos quais os processos de negócio dependem; 3) identificar níveis de serviço e agentes que meçam os serviços; 4) negociar e articular um acordo de níveis de serviço (*Service Level*

*Agreement*, ou SLA); 5) produzir relatórios de níveis de serviço e compará-los com os SLAs; e 6) realizar ajustes finos na empresa para melhorar cada vez mais os serviços.

Um parâmetro de serviço é uma variável cujo valor permite avaliar o desempenho de um serviço. Um nível de serviço é um valor de um parâmetro de serviço utilizado para negociar uma qualidade de serviço desejável. Um acordo de níveis de serviço (SLA) é um contrato entre um fornecedor e um consumidor que identifica: 1) serviços suportados pela rede da empresa; 2) parâmetros de serviço para cada serviço; 3) níveis de serviço; e 4) obrigações do fornecedor e consumidor quando os níveis de serviço não são cumpridos. Um relatório de níveis de serviço (*Service Level Report*, ou SLR) é um documento que mostra o SLA e a evolução verificada de um parâmetro de serviço ao longo de um certo período. A figura 2.14 ilustra os conceitos mais importantes em SLM e as relações entre eles.

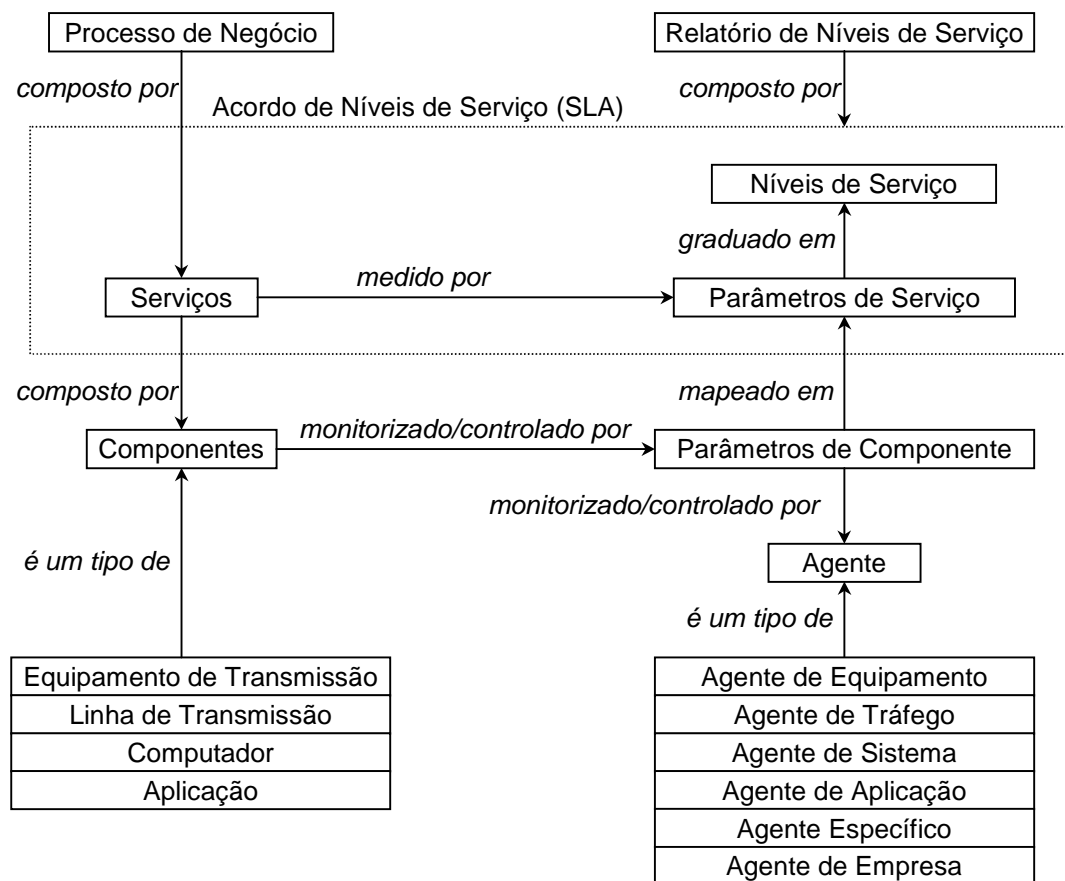


Figura 2.14: Conceitos envolvidos em SLM.

A metodologia de SLM divide-se, grosso modo, em três fases [Lewis 99a]:

- **Fase 1: Requisitos e análise.** Esta fase corresponde a um estudo preliminar, podendo ser dividida em:
  - Passo 1. O fornecedor e o consumidor trabalham em conjunto para chegar à compreensão mútua dos processos de negócio do consumidor.
  - Passo 2. O fornecedor e o consumidor trabalham em conjunto para chegar à compreensão mútua dos serviços necessários aos processos de negócio.
  - Passo 3. O fornecedor e o consumidor trabalham em conjunto para chegar à compreensão mútua dos parâmetros dos serviços e dos níveis de serviço para cada serviço.
- **Fase 2: Desenho, teste e integração.** Esta fase corresponde à adaptação ao mundo real, podendo ser dividida em:
  - Passo 4. O fornecedor faz um inventário dos componentes: a topologia da rede, os tipos de equipamentos e meios de transmissão, os tipos de sistema e aplicações que são usadas, e os processos de gestão existentes.
  - Passo 5. O fornecedor faz uma primeira aproximação a correlacionar os serviços e os componentes.
  - Passo 6. O fornecedor faz uma primeira aproximação a: 1) seleccionar parâmetros de componentes para medir e (opcionalmente) controlar os componentes, e 2) mapear esses parâmetros nos parâmetros dos serviços.
  - Passo 7. O fornecedor faz uma primeira aproximação a: 1) identificar agentes para monitorizar e controlar os componentes, 2) conceber a integração dos agentes, e 3) experimentar com protótipos de pré-produção.
- **Fase 3: Produção.** Esta fase corresponde à realização prática da SLM, podendo ser dividida em:
  - Passo 8. O fornecedor passa o sistema para produção e estabelece uma referência para produzir o primeiro SLR. Uma referência (*baseline*) [Pozzi 96] é um conjunto de medidas estatísticas feitas num período de tempo que caracteriza a utilização e desempenho da rede.
  - Passo 9. O fornecedor e o consumidor revêem o primeiro SLR, e negociam um primeiro SLA.



- Passo 10. A produção prossegue, sendo os SLRs e SLAs verificados, continuados, ou, opcionalmente, renegociados no fim de cada período, tipicamente um mês.

Em [Decisys 97] é proposto um ciclo de revisão de SLAs ilustrado na figura 2.15. Neste ciclo, depois de definir os objectivos a atingir, monitoriza-se o desempenho, tenta-se identificar os focos de problemas, fazem-se modificações à rede para tentar resolver os problemas identificados (por exemplo: afinar parâmetros, reconfigurar, actualizar componentes da rede), reavaliam-se os objectivos de níveis de serviço, e recomeça-se o ciclo.

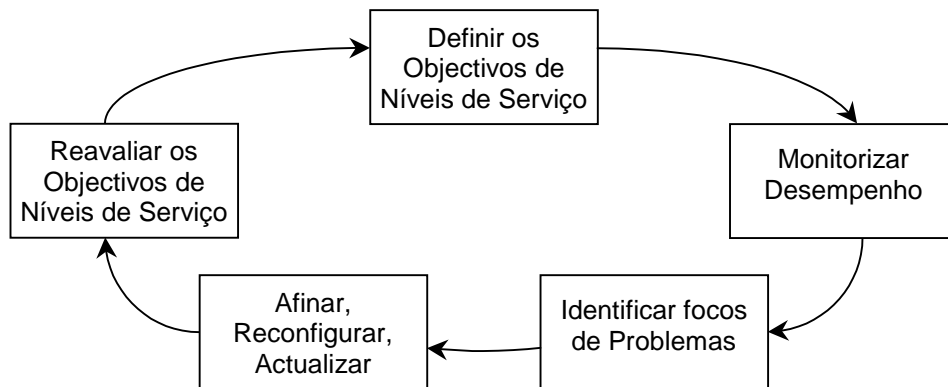


Figura 2.15: Ciclo de revisão de um SLA.

Também na rede telefónica se utilizam alguns conceitos semelhantes aos de SLM. Em [E.801] são apresentadas algumas recomendações para a elaboração de SLAs e SLRs, embora aqui sejam designados *Service Quality Agreements*.

O TeleManagement Forum (TM Forum), um consórcio internacional de fornecedores de serviço, está a trabalhar no sentido de desenvolver normas na área de SLM. Em [GB 917] é proposto um ciclo de vida para um serviço e para o seu SLA, conforme ilustrado na figura 2.16. O desenvolvimento de novos serviços é normalmente despoletado pelas necessidades dos clientes, ou pela competição. Quando o serviço é vendido, pode ser negociado um SLA com o cliente. Depois, o serviço é implementado, reservando-se os recursos, configurando o novo cliente, e activando o serviço. A execução corresponde ao funcionamento normal do serviço, sendo produzidos relatórios dos SLAs, e resolvidos problemas em tempo real. Por fim, a fase de avaliação

corresponde a analisar a QoS obtida pelos clientes, e a rever os objectivos do fornecedor de serviço.

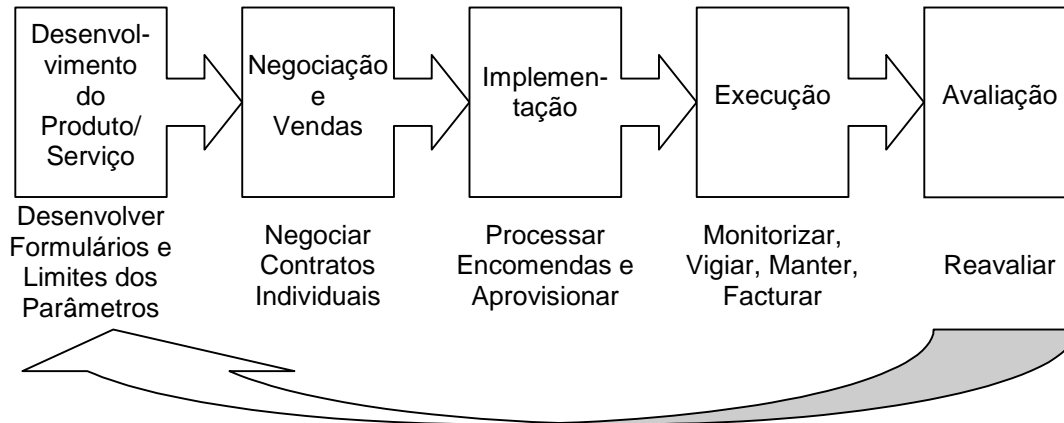


Figura 2.16: Ciclo de vida de um serviço e do seu SLA.

Para a realização prática de SLM é necessário ter uma arquitectura que realize a monitorização dos vários componentes envolvidos nos serviços, tal como ilustrado anteriormente na figura 2.14. Uma extensão mais ambiciosa da SLM corresponde a automatizar alguma da gestão, sendo uma forma de o fazer recorrer a políticas de gestão [Lewis 99d], que serão apresentadas com maior detalhe na secção seguinte. Nesta tese pretende-se seguir este caminho de colaboração entre a área de políticas e a de SLM. Na figura 2.17 ilustra-se um exemplo de uma arquitectura SLM para gestão de falhas. Cada um dos três níveis reage numa escala de tempo diferente. Em cada nível há um primeiro passo de monitorização e interpretação dos eventos, seguido da aplicação de regras de inferência que decidem que acções tomar, e, finalmente, no último passo executam-se as acções decorrentes dessas decisões no bloco de controlo. O nível mais baixo pode corresponder a filtragem de eventos, criação de alarmes, ou ajustes de parâmetros. O nível mais alto normalmente corresponde a decisões conjuntas do fornecedor e do consumidor do serviço para melhorar o cumprimento dos SLAs.

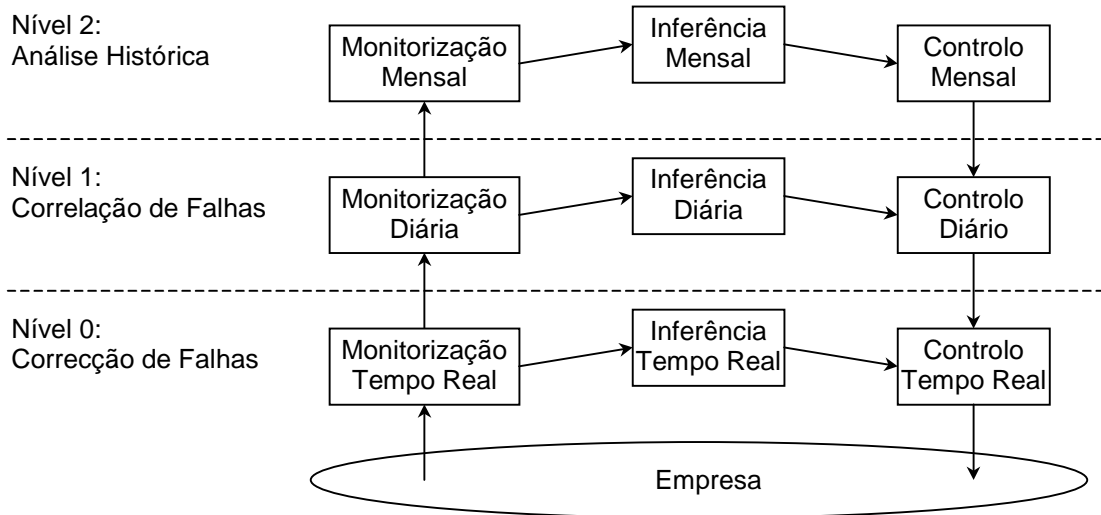


Figura 2.17: Exemplo de gestão de falhas numa arquitectura SLM.

Como exemplo prático de SLAs, apresenta-se o caso do fornecedor de serviço Internet UUNet [UUNET] que oferece garantias de qualidade de serviço aos seus clientes sob a forma de SLAs. Com pequenas variações consoante a localização e tipo de serviço, são oferecidas:

- **Garantia de latência da rede**, especificando que o tempo de atraso médio mensal, de ida e volta, dentro da rede da UUNet deve ser inferior a 120 ms dentro dos EUA, ou dentro da Europa, e inferior a 300 ms dos EUA (Nova Iorque) para a Europa (Londres). Se a garantia falhar em algum dos limites estabelecidos, o fornecedor de serviço credita ao cliente o valor equivalente ao custo do acesso por um dia.
- **Garantia de entrega de pacotes**, especificando que a entrega de pacotes num mês deve ser superior a 99% entre nós dentro dos EUA, ou dentro da Europa, ou dos EUA para a Europa. Se a garantia falhar em algum dos limites estabelecidos, o fornecedor de serviço credita ao cliente o valor equivalente ao custo do acesso por um dia.
- **Garantia de disponibilidade do serviço**, especificando que deve ser de 100%, excluindo a manutenção programada com pelo menos 48 horas de antecedência. Se a garantia falhar, o fornecedor de serviço credita ao cliente, por cada hora ou fracção de indisponibilidade, o valor equivalente ao custo do acesso por um dia.
- **Garantia de reporte de falhas**, especificando que o cliente deve ser notificado pelo fornecedor de serviço de falhas no prazo de 15 minutos. O fornecedor de serviço faz

um *ping*<sup>4</sup> ao *router* do cliente a cada 5 minutos, e notifica-o se dois *pings* consecutivos falharem. Se a garantia falhar, o fornecedor de serviço credita ao cliente o valor equivalente ao custo do acesso por um dia, no máximo uma vez por dia.

- **Garantia de instalação de circuito**, especificando que uma linha deve ser instalada dentro de 40 dias úteis para linhas Frame Relay, 56 Kbps, ou T1 (1544 Kbps), 60 dias úteis para linhas T3 (44,736 Mbps), ou o prazo acordado para os restantes tipos de linha. Se a garantia falhar, o fornecedor de serviço credita ao cliente 50% do valor correspondente à instalação.

Nos casos de serviços de Rede Privada Virtual (*Virtual Private Network*, ou VPN), as garantias de latência e disponibilidade de serviço só são aplicáveis se o nível de utilização sustentado das linhas for menor ou igual a 50% da capacidade instalada. Se se exceder este limite, o fornecedor de serviço notifica o cliente, e este tem 30 dias para pedir o aumento de capacidade, caso queira continuar a ter estas garantias válidas.

Verifica-se que há grandes semelhanças entre os conceitos de SLM e os de QoS apresentados na secção 2.6. Há, no entanto, algumas pequenas diferenças, sendo talvez a mais importante a relativa ao contexto em que ambas são normalmente aplicadas. Enquanto a QoS é normalmente referida a redes de telecomunicações e aplicações multimédia, a SLM refere-se habitualmente a estruturas de redes empresariais. Assim, os serviços em SLM referem-se normalmente a serviços de utilidade para os clientes da empresa ou os utilizadores dos sistemas da empresa, correspondendo a um grau de abstracção mais alto do que os serviços proporcionados directamente pelos componentes da rede.

De uma perspectiva de negócio, a SLM corresponde a **gestão de riscos** [Bhoj 01]. Na realidade, o fornecedor do serviço tem de pesar os benefícios financeiros de oferecer SLAs aos seus clientes, contra o risco de perda financeira, directa ou indirecta, como resultado da violação de SLAs. De forma semelhante, as companhias de seguros pesam a probabilidade de ocorrer um desastre nos prémios que cobram aos seus clientes. Também os bancos que emitem cartões de crédito incluem os prejuízos decorrentes das fraudes nas comissões que cobram aos comerciantes. Em ambos os

---

<sup>4</sup> corresponde ao envio de um ICMP (*Internet Control Message Protocol*) [RFC 792] Echo e aguardar um ICMP Echo Reply.

casos, as empresas não controlam directamente os desastres ou fraudes. Assim, os fornecedores de SLM necessitam de dados históricos para avaliar o compromisso entre oferecer um nível elevado de serviço e as penalidades por violação de SLAs.

O risco de violação de SLAs pode ser reduzido com o uso de períodos de verificação longos, da ordem de um mês, como no caso da rede UUNet, referido anteriormente. No entanto, apesar da importância dos dados históricos, num mundo competitivo como o actual, a simples monitorização pode não ser suficiente. No caso da rede UUNet, atendendo à cláusula que exige o aumento da capacidade, está a aliar-se a monitorização a um sobredimensionamento da rede, como forma de fazer cumprir os SLAs. Se bem que seja uma solução possível, não é a técnica mais eficiente, para além de causar incómodos aos clientes.

Este problema, de como fazer cumprir os SLAs, raramente é abordado pelos autores que falam de SLM. Em [Sturm 00] são apresentados e analisados alguns produtos para SLM. Contudo, na maior parte dos casos, estes produtos SLM limitam-se a fazer monitorização e relatórios dos níveis de desempenho dos serviços, cabendo a alguém responsável pela gestão da rede analisá-los, e decidir quais os melhoramentos da rede necessários para continuar a assegurar os níveis de desempenho acordados nos SLAs. Procurou-se nesta tese estudar formas de automatizar tarefas de gestão com o objectivo de fazer cumprir os SLAs. Para isso são importantes os conceitos de políticas de gestão, retomando-se este tema na secção seguinte.

## **2.8 POLÍTICAS DE GESTÃO**

Nesta secção aprofunda-se o tema de políticas de gestão já introduzido na secção 2.2.3.2. Começa-se por introduzir várias áreas de aplicação das políticas. De seguida, referem-se formas de classificar políticas, organizar políticas numa hierarquia, refinar políticas, e analisar conflitos entre políticas. Por fim, apresenta-se um exemplo de uma arquitectura de políticas, e discutem-se as vantagens e desvantagens do uso de políticas de gestão.

### 2.8.1 Aplicações das Políticas

As políticas têm uma vasta gama de aplicações desde gestão de serviços em redes, segurança, e modelação de empresas. O conceito de políticas, comum a todas as comunidades, define que as políticas são um conjunto de regras governando as escolhas no comportamento dos sistemas. A motivação para a utilização de políticas é ter um meio flexível de modificar a estratégia de gestão ou o comportamento dos sistemas, sem ter de os modificar ou interromper o seu funcionamento.

Na comunidade Internet há considerável interesse em redes baseadas em políticas. Muito deste trabalho está concentrado em políticas para gestão de QoS em redes, estando o IETF a trabalhar activamente na definição de normas nesta área. Um primeiro modelo genérico de políticas é proposto em [RFC 3060], esperando-se que seja posteriormente estendido para áreas específicas de aplicação.

No domínio da segurança, o trabalho tem sido focado na especificação e análise de políticas de controlo de acessos que evoluíram para controlo de acesso baseado em papéis (*Role-Based Access Control*, ou RBAC) [Lupu 95][Lupu 97b]. Os papéis correspondem a posições dentro de uma empresa às quais estão associados conjuntos de direitos, deveres, funções e interações, resultando na definição de um conjunto de políticas para cada papel [Lupu 97a][Lupu 98].

Alguns grupos de investigação têm utilizado políticas de alto nível para modelação de empresas. O grupo de trabalho da ISO em processamento distribuído aberto (ODP) está a definir conceitos de políticas e papéis na perspectiva de empresa [X.911draft]. Os objectivos da empresa e os SLAs podem ser considerados políticas abstractas de alto nível que devem ser progressivamente refinadas em políticas realizáveis na prática.

### 2.8.2 Classificação, Hierarquia e Refinamento

As políticas podem ser classificadas de acordo com diferentes critérios conforme se ilustra no diagrama multidimensional da figura 2.18 [Wies 94b]. Os vectores da figura representam as dimensões de uma política. Por exemplo, o vector “área funcional” classifica a política no que diz respeito às áreas tradicionais de gestão de redes introduzidas na secção 2.2.2. O vector “modo” classifica a política consoante obrigue, permita ou proíba acções.

Os critérios de classificação são úteis ao processo de refinamento de políticas. No entanto, o significado de cada dimensão na figura 2.18 depende do tipo de objecto gerido pela política e da área funcional. Além disso, os critérios de classificação não são perfeitamente ortogonais, podendo haver interdependências entre eles.

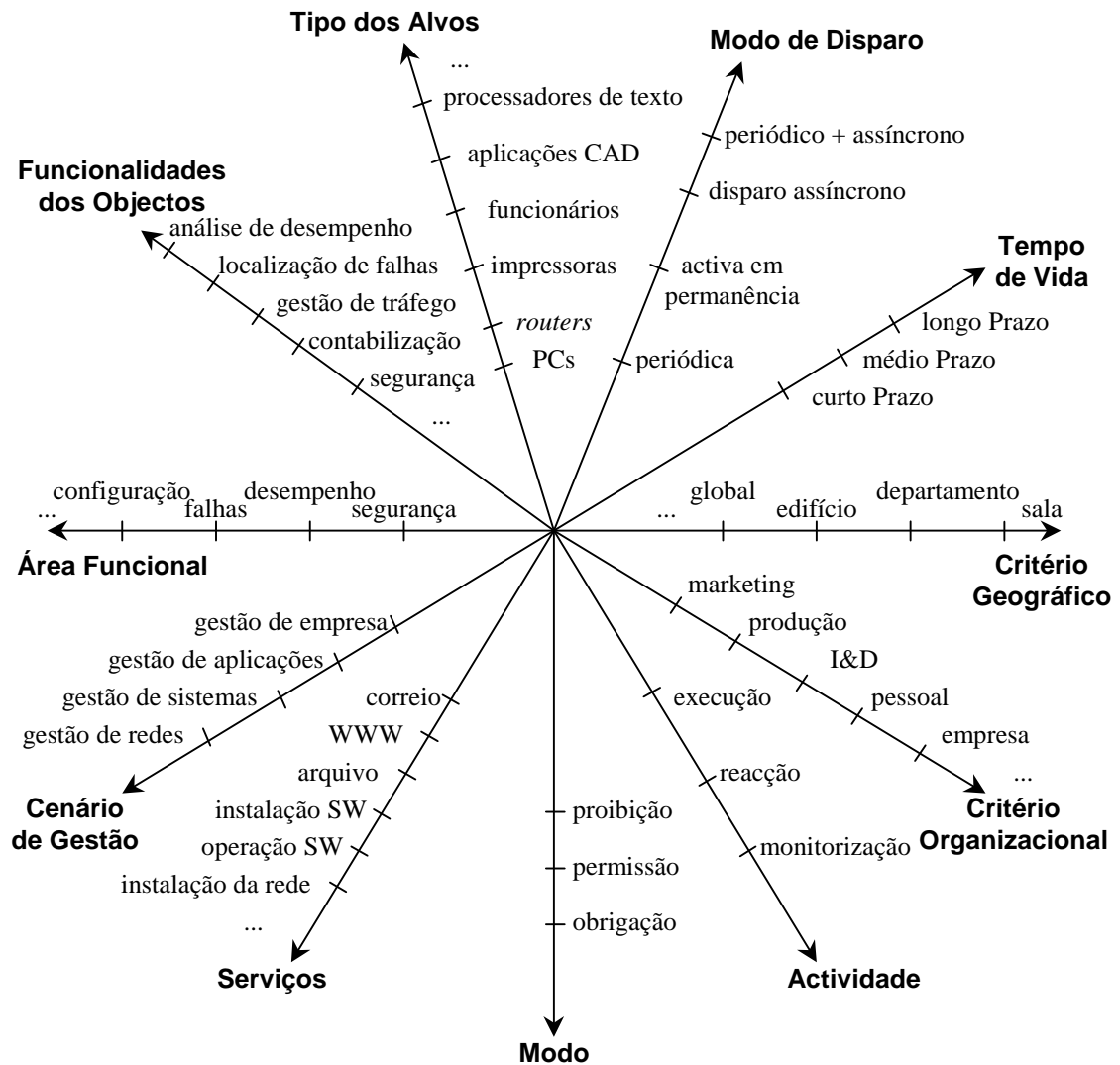


Figura 2.18: Critérios para classificação de políticas.

As políticas podem ser organizadas numa hierarquia consoante o grau de abstracção correspondente, conforme ilustrado na figura 2.19 [Wies 94b]. As políticas corporativas, ou de alto nível, correspondem directamente aos objectivos da empresa, pelo que incluem aspectos de gestão estratégica do negócio, em vez de aspectos de gestão relativos à tecnologia. Para que possam ser aplicadas num ambiente de gestão, as políticas devem ser refinadas para um dos níveis inferiores. As políticas orientadas a

tarefas correspondem a gestão de processos ou tarefas, definindo a forma como as ferramentas de gestão devem ser aplicadas para conseguir os comportamentos desejados nos recursos geridos. As políticas funcionais operam ao nível das funções de gestão de sistema (apresentadas na secção 2.2.3), definindo a sua utilização. As políticas de baixo nível operam ao nível dos objectos geridos, correspondentes aos recursos da rede e sistemas.

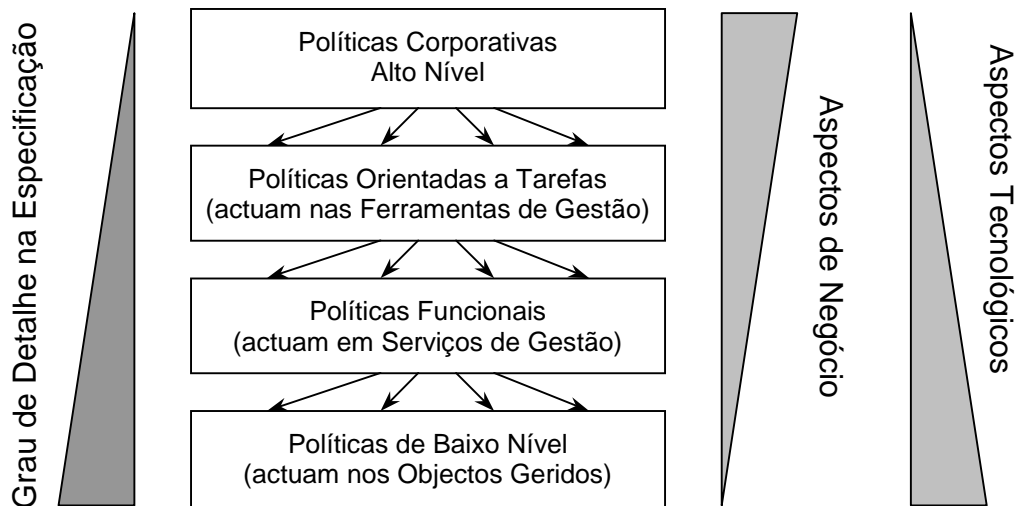


Figura 2.19: Arquitectura hierárquica de políticas.

Os diferentes graus de abstracção ajudam tanto a especificação como a implementação da gestão. O número e a definição dos graus de abstracção varia na literatura, consoante o autor e o critério utilizado para a divisão em níveis que pode ser mais geográfica ou funcional. A arquitectura de camadas lógicas de TMN [M.3010] propõe quatro camadas, tal como descrito anteriormente na secção 1.2 e figura 1.1, e que se recorda aqui: camada de negócio, responsável por toda a empresa (i.e. todos os serviços e redes), que executa a coordenação global dos negócios; camada de serviço, responsável pelos serviços oferecidos por uma ou mais redes; camada de rede, responsável por cada rede; e camada de elemento de rede, que faz a gestão de elementos de rede individuais. Em [Moffett 93b] são propostos três níveis: nível objectivos, especificando objectivos; nível políticas, definindo regras; e nível planos, correspondendo a procedimentos de acções. A mesma notação [Marriott 96] é utilizada em todos os graus de abstracção. [Koch 96] também propõe três níveis: nível requisitos, descritos em prosa; nível de objectivos, utilizando grelhas padrão de atributos



especificando eventos, restrições e acções; e um nível operacional onde são utilizadas uma Linguagem de Descrição de Políticas e uma Linguagem de Definição de Eventos.

O refinamento de políticas pode ser feito de três formas [Moffett 93b]:

- **Decompondo** o objecto gerido nos objectos geridos que o constituem. Por exemplo, a política “Os dados da empresa devem ser protegidos” pode ser decomposta em políticas por cada departamento da empresa para proteger os dados respectivos.
- **Delegando** a responsabilidades de execução das políticas em outras entidades. Por exemplo, a política anterior podia resultar em: “o gestor do departamento X deve proteger os dados do departamento X”.
- **Refinando** os objectivos de gestão para outros mais precisos, ou em acções, ou sequências de acções a realizar. Não há forma automática de refinar os objectivos, tudo dependendo de cada caso particular. A política anterior podia ser refinada em: “o gestor do departamento X deve fazer cópias de segurança semanais dos dados”, e “os utilizadores do departamento X devem usar *login* seguro”.

Em [Apêndice C de X.906draft] são dados exemplos de decomposição de requisitos de QoS com composições sequenciais e paralelas de objectos. Por exemplo, o atraso numa sequência de objectos é igual à soma dos atrasos nos vários objectos individuais. O atraso numa composição paralela de objectos é igual ao atraso no objecto particular seleccionado para a interacção.

### 2.8.3 Conflitos entre Políticas

Um possível problema que pode ocorrer no processo de refinamento é haver **conflitos** entre políticas, isto é, as políticas gerarem contradições. A figura 2.20 [Moffett 93a] ilustra uma classificação dos tipos de conflitos para políticas que têm apenas modos obrigatório ou permitido, e positivo ou negativo.

Os conflitos de modo podem ser detectados sem conhecimento do significado dos objectivos da política. Por exemplo, uma política de obrigação pode definir uma actividade que seja proibida por uma política de autorização negativa.

Por outro lado, os conflitos de objectivos, correspondem a sobreposições de gestores e/ou objectos geridos, dependendo da semântica dos objectivos, e por isso da aplicação particular. Um exemplo de conflito de interesses do domínio bancário é um mesmo gestor dar conselhos a duas entidades. Concretizando: se o gestor estiver a

realizar uma operação de compra de uma empresa para um cliente, e der conselhos de investimento a outro cliente baseado nessa informação, há um conflito, que neste caso é proibido. Em outros domínios de problemas, uma tal utilização de informação pode ser permitida e muito benéfica. Um exemplo de conflito de deveres é o mesmo gestor autorizar um pagamento e assinar o cheque do pagamento, o que viola o princípio de separação de deveres. Outros exemplos de conflitos, incluindo papéis são dados em [Lupu 99].

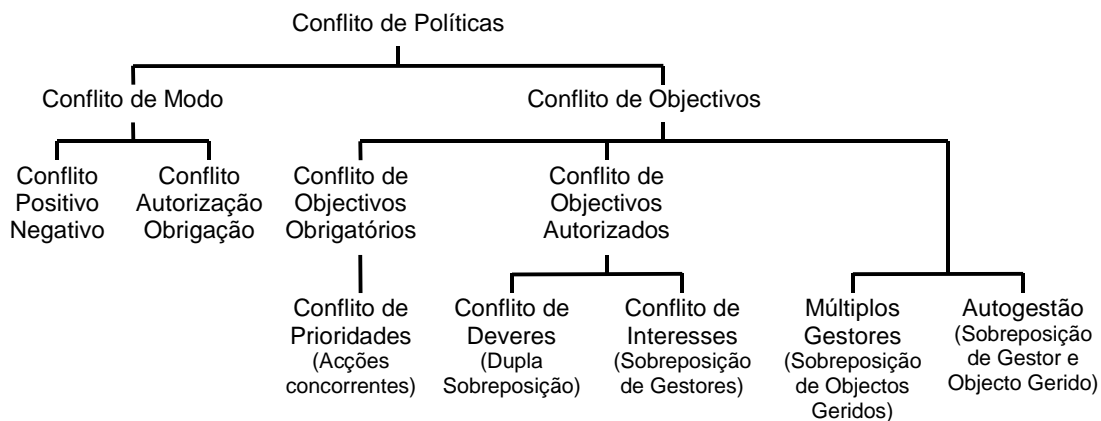


Figura 2.20: Classificação de conflitos entre políticas.

Para evitar, ou resolver conflitos, podem ser tomadas algumas medidas. Por exemplo, os conflitos podem ser detectados por um compilador, ou outro sistema de verificação automática, antes da sua utilização num sistema real. Uma outra opção é utilizar uma linguagem que evite a ocorrência de conflitos não resolvidos, por exemplo através da definição de prioridades. As prioridades podem ser explicitamente indicadas nas políticas, ou podem ser implícitas, por exemplo definindo políticas negativas com maior prioridade, ou dando maior prioridade a políticas mais próximas dos objectos a que se referem.

A detecção de conflitos específicos da aplicação pode ser feita por metapolíticas [Lupu 99], isto é, políticas sobre políticas de gestão. Por exemplo: não pode haver nenhuma política que autorize um gestor a retirar uma política em que ele seja o objecto.

Os conflitos num sistema real podem ser detectados antecipadamente e evitados. Por exemplo, uma forma de conflito de deveres pode ser evitada restringindo o domínio de gestores de uma política para um gestor de cada vez.

Os conflitos entre acções podem ser detectados quando ocorrem. Por exemplo, detectando que foi realizada uma acção que inverte uma acção anterior de outro gestor. Neste caso, pode depender da aplicação em causa se a acção deve ser impedida, ou se deve ser emitido um aviso.

#### 2.8.4 Exemplo de Arquitectura de Políticas

Um exemplo de arquitectura de políticas para controlo de admissão de ligações é definido em [RFC 2753], conforme ilustrado na figura 2.21. Os servidores de políticas contêm pontos de decisão de políticas (*Policy Decision Points*, ou PDP), onde as decisões sobre políticas são tomadas. Os nós da rede (normalmente *routers*) contêm pontos de execução de políticas (*Policy Enforcement Points*, ou PEP), que recebem as políticas através do protocolo COPS [RFC 2748] dos PDPs e as executam. O PDP pode, por sua vez, contactar outros servidores, por exemplo para aceder a bases de dados de informação de configuração, autenticação de utilizadores, contabilização, ou taxação. Há a possibilidade de o PDP e o PEP estarem no mesmo nó da rede.

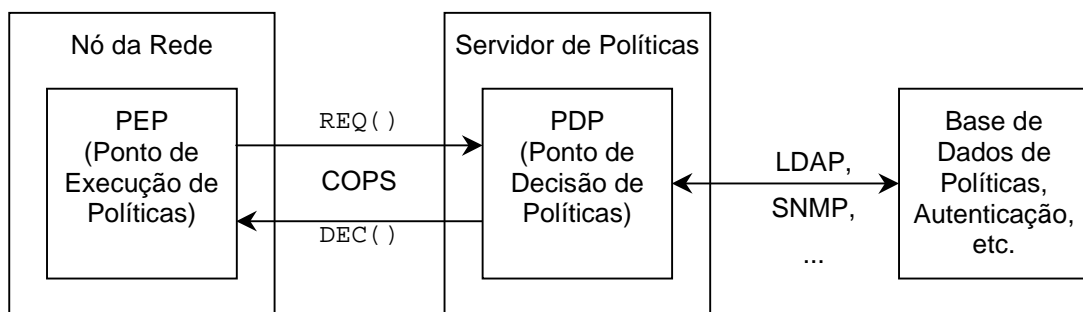


Figura 2.21: Arquitectura de controlo de admissão baseado em políticas.

Há dois modelos previstos para as interacções entre o PEP e o PDP:

- no modelo de delegação (*outsourcing*), o PEP delega responsabilidade de todas as decisões no PDP, havendo um pedido e uma resposta por cada decisão de controlo de admissão.
- no modelo de configuração, ou de aprovisionamento [RFC 3084], por outro lado, o PDP carrega as políticas no PEP. Neste caso, a comunicação entre ambos é mais

rara, sendo apenas necessária quando há modificações de políticas, ou modificações da configuração do equipamento do nó da rede.

O modelo de informação para a definição de políticas [RFC 3060] é constituído por uma hierarquia de classes extensível, que exprime a informação contida nas políticas, as relações entre políticas, e como se controlam as políticas, mas omitindo os detalhes de implementação. As políticas são constituídas por regras que têm uma forma “Se condição, então acção”, com uma estrutura tal como ilustrado na figura 2.22. Uma condição de uma regra é uma disjunção, ou conjunção de condições, possivelmente negadas. Só se a condição de uma regra for satisfeita (i.e., avaliada a verdade) é que a acção é executada. Uma regra também pode ter períodos temporais associados, indicando quando está activa e inactiva. Um conjunto de propriedades permite indicar, entre outras coisas: o estado da política (activa, inactiva, activa para depuração de erros); a prioridade da política; informação sobre a forma de utilização da política; e se as acções devem ser obrigatoriamente executadas, ou não (política de melhor esforço).

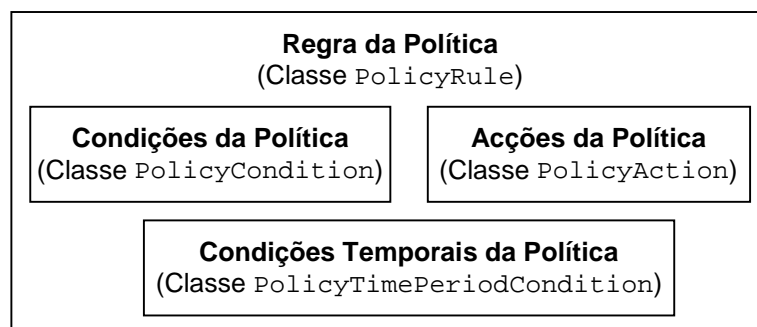


Figura 2.22: Classe que define uma regra de uma política.

### 2.8.5 Discussão

Do exposto nesta subsecção podem-se tirar algumas conclusões. As políticas são regras que representam objectivos de gestão. As políticas de alto nível são em tudo semelhantes aos objectivos de níveis de serviço de SLM, embora os conceitos provenham de comunidades diferentes. As políticas de baixo nível (as que podem ser implementadas), são normalmente regras, que especificam acções a realizar sobre determinados objectos geridos, quando certas condições particulares ocorrem. Estas

regras normalmente são específicas a problemas particulares, tais como: controlo de admissão de ligações; autorização de acesso a recursos; e configuração de componentes. Devido à grande diferença semântica entre os objectivos de alto nível e o que pode ser realizado nos objectos geridos, o refinamento das políticas de alto nível em políticas de baixo nível normalmente não pode ser automatizado, sendo necessário que alguém especializado analise cada caso particular. Além disso, devido à especificidade das políticas de baixo nível, os objectos geridos têm de ser construídos de forma a suportarem as operações pretendidas.

Apesar destas limitações, os conceitos de políticas são de extrema importância para qualquer sistema de gestão moderno, devido à flexibilidade de reconfiguração subjacente.

## 2.9 LINGUAGENS DE GESTÃO

As linguagens mais simples para programação de aplicações de gestão são interfaces de programação (*Application Programming Interfaces*, ou API) para o protocolo de gestão [Pereira 99a]. Exemplos deste tipos de linguagens são:

- a Systems Management Scripting Language (SMSL) [X.753], norma internacional do ITU-T, referida anteriormente na secção 2.2.3.3.
- o Scotty [Schonwalder 95], que é uma extensão à linguagem Tcl (*Tool Command Language*) [Ousterhout 94] para aplicações de gestão, cobrindo entre outros protocolos o SNMP e o CMIP.
- as Java Management Extensions [JMX], para a linguagem Java.
- a NET-SNMP [NET-SNMP], com uma biblioteca para Unix e MS Windows.
- a SNMP++ [Banker 95], sob a forma de classes para C++.
- a WinSNMP [WinSNMP], específica para ambientes MS Windows.

Outras linguagens podem ser construídas sobre estas, como a linguagem DEAL [Znaty 96], baseada em SQL, que permite manipular tabelas para além dos procedimentos normais de gestão. Esta linguagem é traduzida para um *script* SNMP baseado em Tcl, que por sua vez é interpretado pelo Scotty. [Kalyanasundaram 97] apresenta um ambiente de folhas de cálculo onde cada célula contém informação de gestão recolhida e as regras para a recolher, sendo proposta uma linguagem que inclui a

especificação da recolha de informação e de eventos, além dos procedimentos normais de gestão. [Cheng 97] propõe uma linguagem baseada em CCS (*Calculus of Communicating Systems*) para descrever operações de monitorização e controlo. [Brites 94] propõe uma camada de Regras Sintácticas Básicas por cima de uma infraestrutura SNMP, que pode ser estendida com módulos especializados para implementar tipos específicos de aplicações de gestão. As extensões podem incluir eventos, acções e definições de procedimentos tais como recolha de informação, definição de limiares de actuação, registo de informação, e geração de alarmes. [Anerousis 98] e [Stadler 96] propõem modelos para derivar informação das MIBs utilizando expressões que são automaticamente calculadas para fornecer uma visão actualizada do estado dos equipamentos. Actualmente já começam a aparecer normas com algumas funcionalidades deste tipo. Em [RFC 2982] é definida a *expression* MIB, que permite disponibilizar valores calculados por expressões de outros elementos da MIB. Em [RFC 2981] é definida a *event* MIB, que permite estabelecer limiares sobre variáveis de MIBs locais, ou remotas, com o objectivo de desencadear o envio de notificações, ou a atribuição de novos valores a variáveis da MIB. Em [RFC 3231] é definida a *schedule* MIB, que permite programar a atribuição de novos valores a variáveis da MIB periodicamente, ou em datas e horas especificadas.

Outros métodos de mais alto nível de automatização de operações de gestão são habitualmente baseados em técnicas de inteligência artificial, inteligência artificial distribuída, políticas, ou papéis. No entanto, políticas e papéis são mais adequados para gestão de segurança, configuração e contabilização, sendo pouco adequadas para gestão de desempenho, falhas e planeamento, às quais foi dada ênfase nesta tese.

## 2.10 DISCUSSÃO E MOTIVAÇÃO PARA A TESE

Para assegurar os requisitos de qualidade dos processos de negócio de uma empresa é importante utilizar não só os conceitos de gestão de níveis de serviço (SLM) apresentados na secção 2.7, como também os conceitos de qualidade de serviço (QoS) apresentados na secção 2.6.

Como se disse anteriormente, actualmente a SLM quase só faz monitorização e relatórios, e as situações em que actualmente se assegura QoS são quase só a nível de rede e aplicações multimédia.

Uma forma que se pensa ser adequada para se poder assegurar requisitos de QoS em múltiplos níveis de abstracção é recorrer aos conceitos de políticas de gestão apresentados na secção 2.8. No entanto, a maior parte das utilizações de políticas actuais são situações de configuração e autorização, onde as políticas têm um papel passivo.

Assim, nesta tese propõe-se a utilização de políticas activas de gestão que monitorizam continuamente a rede e actuam em tempo real no sentido de assegurar a QoS pretendida.





# CAPÍTULO 3

## MODELO DE GESTÃO

Neste capítulo apresenta-se a proposta desta tese, que consiste num modelo de gestão baseado numa hierarquia de políticas activas, com vários níveis de abstracção, para assegurar os requisitos de gestão pretendidos. Este modelo foi inicialmente apresentado em [Pereira 99b]. Descreve-se ainda a metodologia de análise de problemas de gestão e de construção de políticas activas. Dão-se exemplos de regras de actuação que podem facilitar a construção das políticas activas. Descreve-se a arquitectura para a implementação das políticas activas. Finalmente, conclui-se o capítulo descrevendo as vantagens da abordagem proposta.

### 3.1 DESCRIÇÃO GERAL

A principal tarefa do gestor de uma rede é oferecer aos utilizadores finais o melhor serviço possível com custo mínimo. À medida que as redes crescem em número, dimensão, complexidade, dinamismo e variedade de equipamento, as tarefas de gestão e controlo tornam-se cada vez menos triviais, tornando significativo o esforço que é necessário dedicar à gestão da rede e dos seus sistemas.

As tarefas do gestor da rede podem ser simplificadas automatizando os procedimentos simples, ou utilizando ferramentas que lhe permitam tomar decisões de alto nível, executando operações suficientemente poderosas para atingir os objectivos de gestão.

A arquitectura de gestão proposta nesta tese é baseada numa hierarquia de políticas activas de gestão, funcionando em diferentes níveis de abstracção. Cada **política activa** automatiza as operações de gestão relacionadas com um certo aspecto de gestão, conforme especificado num contrato feito com o utilizador que lançou a política activa. Os **contratos** são definições de configuração gerais, que especificam as garantias de serviço necessárias para atingir os objectivos de gestão. Estes contratos podem

corresponder a acordos de níveis de serviço (*Service Level Agreements*, ou SLAs), ou a acordos em outros níveis de abstracção. É possível monitorizar o cumprimento das garantias através de parâmetros de qualidade de serviço (QoS) de gestão que oferecem uma medida de como as funções das políticas activas estão a ser desempenhadas, e de como os objectivos de gestão estão a ser atingidos.

### 3.2 NÍVEIS DE ABSTRACÇÃO, HIERARQUIA E REFINAMENTO

Viu-se anteriormente que o conceito de **processos de negócio** é utilizado em empresas modernas para agregar as tarefas necessárias ao cumprimento de um certo objectivo (vendas, contabilidade, manufactura de produtos, etc.). A figura 3.1 ilustra um exemplo de uma situação na qual os processos de negócio de uma empresa típica se suportam em **serviços**, que actualmente são maioritariamente baseados em computadores e em redes. Estes serviços são habitualmente implementados como aplicações distribuídas sobre um ambiente distribuído [Vinoski 02] que oferece independência de localização, invocação remota transparente e notificação de eventos. Cada serviço é composto por algumas **tarefas** que são normalmente executadas em processos a correr nos computadores da empresa. Os computadores comunicam através da **rede**. Tanto a execução de tarefas como a comunicação através da rede estão baseadas em **equipamentos**, tais como: computadores, periféricos e equipamento de rede.

Cada uma das camadas identificadas (processo de negócio, serviços, tarefas, rede e equipamento) corresponde a um nível de abstracção diferente. Contudo, esta divisão não é rígida, podendo algumas camadas não existir, ou, por outro lado, ser decompostas em subcamadas para simplificar a realização do sistema. O TM Forum está a desenvolver um guia [GB 921] que apresenta uma arquitectura de processos de negócio para um fornecedor de serviço. No entanto, os níveis detalhados neste guia correspondem a vários níveis de abstracção dentro do que se considerou a camada de processos de negócio nesta tese.

Os requisitos do nível processo de negócio, normalmente um texto em prosa, devem ser sucessivamente refinados para requisitos que façam sentido em cada nível de abstracção. Poderá ser necessário prever requisitos de disponibilidade, desempenho,

segurança, configuração e contabilização para cada serviço, tarefa, rede e equipamento. Além do refinamento, alguns requisitos específicos à implementação de cada nível terão de ser acrescentados pelo gestor para configurar esse nível, especialmente na passagem entre os níveis de abstracção mais altos. Por exemplo, o número de servidores necessários e a capacidade da rede têm de ser especificados, mesmo que não estejam presentes nos requisitos de alto nível. Todos os requisitos devem ser formalizados como contratos de gestão para as políticas activas que os vão assegurar. Esta parte da hierarquia de níveis de abstracção e relações entre eles tem um papel central no modelo de gestão desta tese.

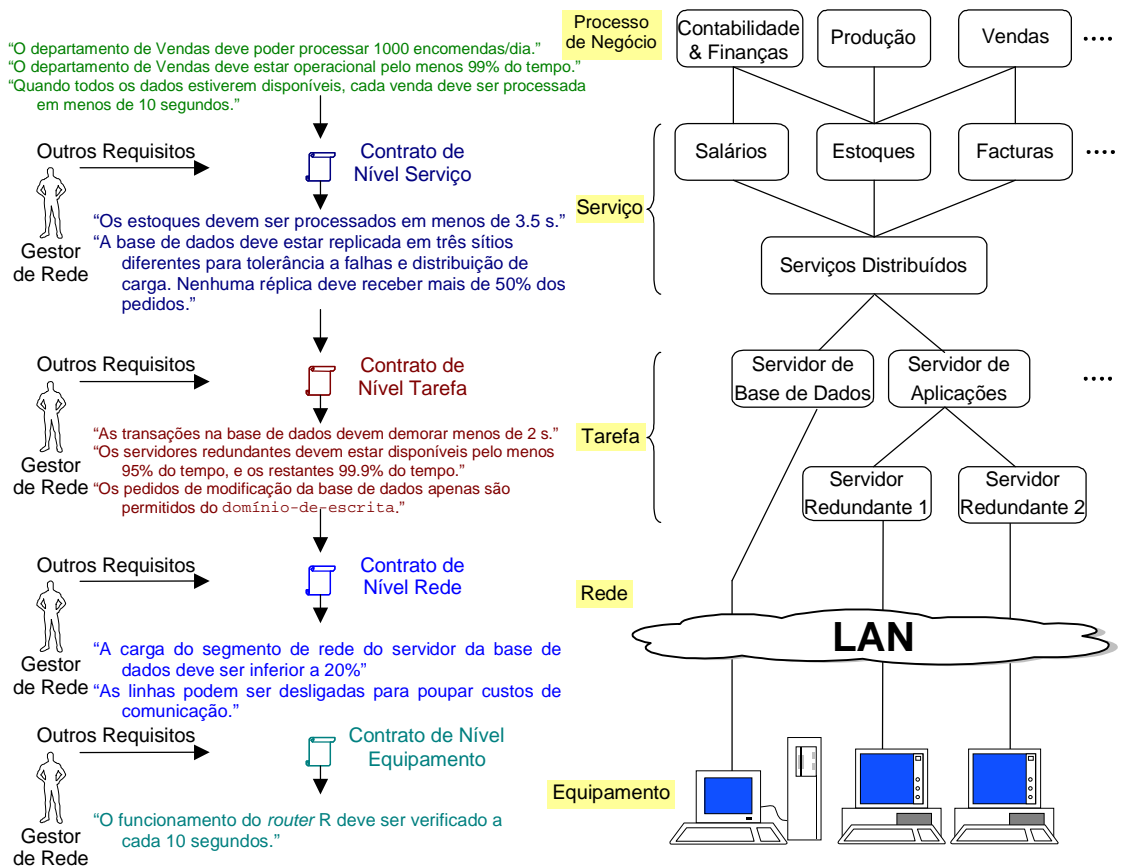


Figura 3.1: Exemplo de níveis de abstracção.

O processo de refinamento deve ser feito analisando as dependências de cada processo de negócio, e mapeando os requisitos de alto nível em parâmetros dos objectos geridos. Tomando como exemplo o caso dos requisitos a impor a uma venda, a figura 3.2 ilustra uma possível decomposição de uma operação de venda. Os diferentes

objectos do sistema são utilizados para estabelecer contratos para as políticas activas de cada nível de abstracção. O correspondente contrato de nível tarefa seria assim:

“As transacções na base de dados devem ser realizadas em menos de 2 segundos”.

“A aplicação deve responder em menos de 2 segundos”.

“A verificação dos dados deve ser feita em menos de 1 segundo”.

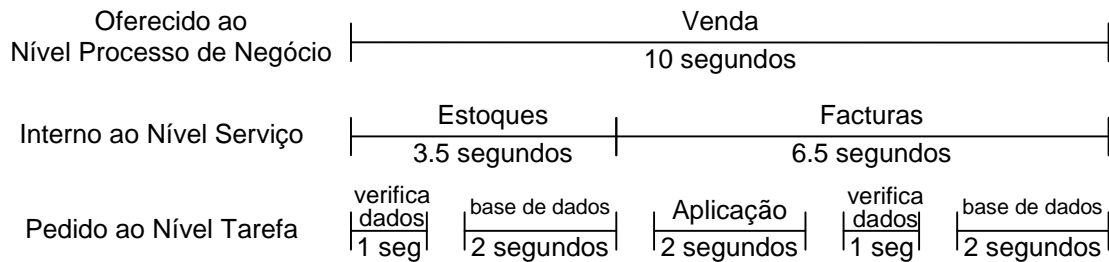


Figura 3.2: Exemplo de refinamento de requisitos de uma venda.

Nos níveis de abstracção mais elevados, o processo de refinamento claramente necessita de intuição humana para definir os requisitos de configuração e o comportamento específico das políticas activas que actuam para assegurar os contratos. Pensa-se que, abaixo de um certo nível de abstracção, o refinamento possa ser feito automaticamente, pela selecção do comportamento correcto de entre as políticas activas disponíveis nesse nível, pelo estabelecimento de contratos e pela selecção dos habituais requisitos de configuração. Na realidade, as políticas activas de médio e baixo nível têm um comportamento genérico que actua sobre objectos geridos reais, tais como servidores, *routers*, troncas, parâmetros de sistema, e instrumentação em geral. Desta forma, as políticas activas actuam autonomamente entre certos valores limites, tal como especificado nos seus contratos, bastando assim parametrizar o seu comportamento genérico.

Uma hierarquia de políticas activas é utilizada para decompor o problema de gestão em diversos subproblemas em diferentes níveis de abstracção e em diferentes partes da rede. Para fazer cumprir os seus contratos, as políticas activas usam o seu próprio esforço e o de políticas activas de níveis inferiores, tal como especificado em contratos feitos com cada uma delas. Estes contratos podem ser dinamicamente modificados pelas políticas activas de nível superior durante o funcionamento do

sistema, para se adaptar a modificações nas condições da rede ou dos sistemas, e melhor cumprir os contratos de alto nível.

À medida que se desce na hierarquia de políticas activas, a rede torna-se mais segmentada. O nível imediatamente acima de uma segmentação pode levar a segmentação em consideração como um grau de liberdade para fazer cumprir o seu contrato. Um exemplo é fazer balanceamento de carga entre diferentes servidores redundantes, tal como ilustrado anteriormente na figura 3.1.

### 3.3 POLÍTICAS ACTIVAS

As **políticas activas** são objectos activos com estado interno e um conjunto de operações, especializados em assegurar certas condições numa parte da rede, resolvendo de forma automática os problemas correspondentes. As condições que uma política activa trata são formalizadas num **contrato** com a entidade que a lançou. O cumprimento dos contratos pode ser monitorizado por parâmetros de qualidade de serviço (QoS) de gestão. A figura 3.3 mostra a estrutura interna das políticas activas para o caso particular das políticas activas de nível serviço.

Para uma política activa assegurar um contrato pode recorrer a contratos com outras políticas activas de níveis mais baixos. Em cada nível, várias políticas activas podem estar a operar para simplificar a implementação, sendo a estrutura interna idêntica para cada uma delas (contratos, requisitos de configuração e parâmetros de QoS). Cada política activa monitoriza os parâmetros de QoS dos níveis inferiores, recebe eventos e invoca acções sobre os objectos geridos sob seu controlo. Internamente, as políticas activas genéricas têm um módulo de monitorização que calcula a QoS, um módulo de reacção para dar uma resposta rápida aos problemas, e uma parte proactiva que actua com base na história dos eventos para ajudar ao planeamento. Um exemplo de monitorização é determinar o atraso dos pacotes extremo a extremo. Um exemplo de gestão reactiva é quando o atraso ultrapassa 80% do limite imposto, modificar o encaminhamento de forma a usar linhas menos carregadas. Um exemplo de gestão proactiva é verificar que actuar quando se ultrapassa 80% do limite imposto não é suficiente, e passar a actuar quando se ultrapassa 75% do limite imposto.

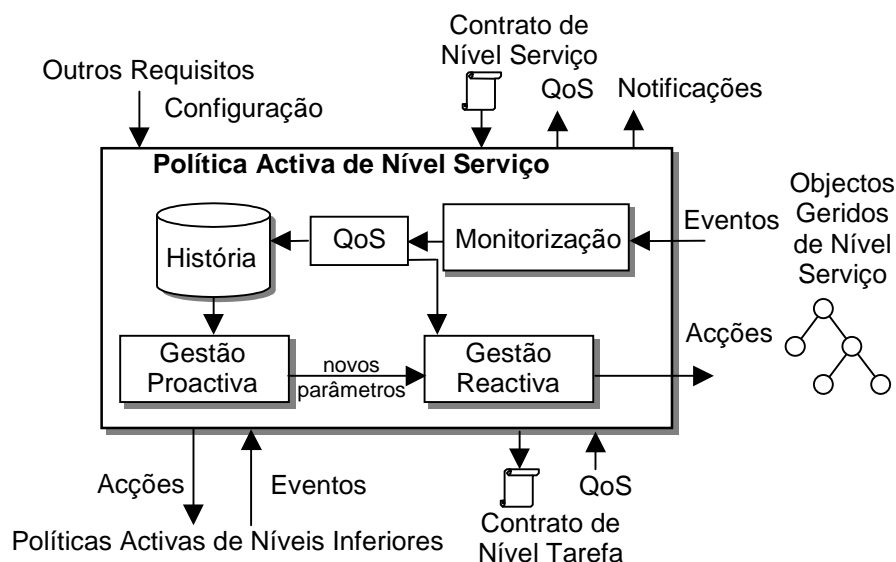


Figura 3.3: Estrutura interna das políticas activas.

Como optimização, para evitar uma monitorização intensiva da QoS, as políticas activas podem enviar notificações de certas situações (eventos) para outras políticas activas, em qualquer nível de abstracção, que estejam interessadas em receber essas notificações. Um efeito suplementar das políticas activas é a descentralização do processamento de informação de gestão e a execução automática de acções, diminuindo o tráfego na rede, aumentando a escalabilidade, e assegurando tempos de resposta mais rápidos, conforme visto anteriormente na secção 2.3.

Nos casos mais habituais, uma política activa apenas envia eventos para outras políticas activas do seu nível de abstracção, numa perspectiva de cooperação para atingir os objectivos, ou para o nível de abstracção imediatamente superior, numa perspectiva de assinalar uma excepção. Além disso, uma política activa apenas actua sobre objectos geridos do seu nível de abstracção, ou sobre políticas activas do nível imediatamente inferior. Constitui-se, assim, uma hierarquia de políticas activas. No entanto, é de prever que, em situações de excepção, ou como resultado de tarefas de planeamento, uma política activa possa actuar directamente sobre qualquer política activa, ou objecto gerido, no seu nível de abstracção, ou num nível inferior. A actuação sobre outra política activa pode ser feita de duas formas, normalmente equivalentes: pela invocação de acções dessa política activa, ou pela modificação do contrato feito com essa política activa. A diferença é o tipo de interface pretendido, que no primeiro caso é uma interface de métodos de objectos, e no segundo caso de parâmetros de contratos.

A aplicação de gestão corresponde à hierarquia de políticas activas, construída de acordo com o processo de refinamento. Desde que o contrato de alto nível esteja a ser respeitado, o gestor deixa de ter de se preocupar com os aspectos tratados pelas políticas, e pode trabalhar a um grau de abstracção mais elevado.

### 3.4 CONTRATOS

Os contratos são construídos com as garantias oferecidas pelas políticas activas. Cada garantia tem uma semântica associada, que pode ser:

- **Melhor esforço:** a política activa faz o seu melhor para cumprir a garantia. Exemplo: o atraso dos pacotes deve ser o menor possível.
- **Estatística:** a política activa permite uma certa fracção de violações da garantia. Exemplo: o atraso dos pacotes extremo a extremo deve ser inferior a 500 ms em pelo menos 95% dos pacotes, contados em cada hora.
- **Determinística:** a política activa assegura sempre a garantia. Exemplo: nenhum pacote demora mais de 1 ms a ser transmitido - a política garante que os dados são segmentados em pacotes suficientemente pequenos para cumprir a exigência.

Esta divisão corresponde a políticas activas que oferecem garantias cada vez mais rígidas, prevendo-se que as penalidades em caso de incumprimento sejam também cada vez maiores.

As definições de requisitos a especificar nos contratos podem ser de um dos seguintes tipos:

- **Quantitativa:** quando são especificados valores numéricos. Exemplo: a taxa de perda de pacotes deve ser inferior a  $10^{-4}$ .
- **Qualitativa:** quando apenas se especifica um indicador do valor. Exemplo: a taxa de perda de pacotes deve ser moderada. Isto pode significar que deve ser inferior a  $10^{-4}$  na maior parte do tempo, podendo ser superior em períodos de congestão, e ser zero em períodos de pouco tráfego. Neste caso, torna-se necessário concretizar o requisito no processo de refinamento.
- **Relativa:** quando se indica uma relação entre dois tipos de serviço. Exemplo: a taxa de perda de pacotes deste serviço deve ser inferior à de qualquer outro serviço.

Classificaram-se os requisitos nestes três tipos de forma a cobrir a maioria das situações práticas.

As condições a assegurar nos contratos também podem ser classificadas consoante os objectivos estabelecidos e consoante o modo de actuação. Os vários tipos de objectivos podem ser:

- **Com Limites:** quando são especificados limites máximos, ou mínimos a impor.
- **Sem Limites:** quando não são especificados limites, pretendendo-se que o valor seja o maior, ou o menor possível.
- **Valor óptimo e desvio:** quando é especificado o valor pretendido, e o desvio máximo tolerado.

Estes três tipos enquadram-se na especificação de requisitos de qualidade de serviço.

O modo de actuação pode ser:

- **Autorização** de fazer algo.
- **Proibição** de fazer algo.
- **Obrigaçã**o de fazer algo.

Esta classificação corresponde à filosofia tradicional de políticas [Sloman 94].

As políticas activas podem ter vários modos de funcionamento, para depuração de erros:

- Modo **passivo:** quando a política activa tem a parte de actuação sobre os objectos geridos desactivada, apenas realizando a monitorização, cálculo dos parâmetros de QoS, e indicando as acções que faria sobre os objectos geridos para análise do seu funcionamento interno, e depuração de erros.
- Modo **activo:** quando a política activa está a funcionar com todas as suas capacidades activadas.

Os contratos podem não ter um resultado de funcionamento tão simples como sucesso ou insucesso. Enquanto funcionam, as políticas activas disponibilizam indicadores do sucesso da sua actividade aos seus utilizadores através de parâmetros de QoS de gestão. Isto permite que o nível superior de políticas activas monitorize o seu funcionamento, de forma a poder actuar a um grau de abstracção mais elevado, com o objectivo de evitar falhas dos contratos dos dois níveis, caso algo não corra de acordo com o plano.



Desta forma, os parâmetros de QoS de gestão são importantes para policiar o modo como uma política activa executa a sua tarefa. Isto é ainda mais importante para garantias com semântica de melhor esforço, que nunca resultam em violações de contratos, mas apenas oferecem uma métrica da distância às condições ideais. Estes parâmetros de QoS de gestão podem ser vistos como uma generalização para gestão de redes dos parâmetros de QoS comuns apresentados na secção 2.6. Será assim natural que os contratos também especifiquem as situações, ou os limiares de funcionamento, em que uma política activa deve actuar, ou enviar uma notificação, para o utilizador com quem fez esse contrato.

### 3.5 METODOLOGIA

Para a construção de sistemas de gestão por políticas activas propõe-se a utilização de uma metodologia baseada na experiência obtida e na metodologia descrita na secção 2.7, devida a [Lewis 99a]. A metodologia tem os seguintes passos:

- **Fase 1: Requisitos e análise.** Esta fase corresponde a um estudo preliminar, podendo ser dividida em:
  - Passo 1. Análise dos processos de negócio existentes e dos requisitos a exigir para cada um.
  - Passo 2. Decomposição dos processos de negócio em vários níveis de abstracção, nomeadamente identificando os serviços, tarefas, redes e equipamentos necessários a cada um.
  - Passo 3. Identificação de parâmetros de QoS em cada nível de abstracção e os níveis de QoS pretendidos para cada parâmetro, através de uma primeira aproximação ao mapeamento dos requisitos dos processos de negócio nos parâmetros de QoS dos diferentes níveis de abstracção.
  - Passo 4. Identificação dos parâmetros a medir e controlar em cada componente de cada nível de abstracção. Primeira tentativa de identificação de um modelo que relacione os parâmetros dos componentes identificados com os parâmetros de QoS, tanto para medir como para controlar a QoS.
- **Fase 2: Desenho, teste e integração.** Esta fase corresponde à adaptação ao mundo real, podendo ser dividida em:

- Passo 5. Análise dos requisitos pretendidos para verificar se são realizáveis, e, se necessário, estudo de alterações à rede e aos sistemas que possam ser feitas para tornar os requisitos realizáveis.
- Passo 6. Escolha da fórmula de cálculo de cada parâmetro de QoS.
- Passo 7. Selecção e configuração das políticas activas necessárias para cada componente em cada nível de abstracção. Pode ser necessário desenvolver novas políticas activas específicas para a situação em causa. Por fim, torna-se necessário configurar o comportamentos dos blocos de gestão reactiva e gestão proactiva das políticas activas.
- Passo 8. Análise de conflitos entre políticas activas, e sua resolução.
- Passo 9. Instalação das políticas activas. Activação das políticas activas em modo passivo para analisar a QoS actual da rede e as formas de actuação propostas pelas políticas activas.
- **Fase 3: Execução.** Esta fase corresponde à realização prática da gestão, podendo ser dividida em:
  - Passo 10. Colocação das políticas activas em modo activo, uma de cada vez. Verificação da estabilidade das políticas activas e validade dos modelos utilizados. Análise dos melhoramentos obtidos na QoS. Ajuste de parâmetros das políticas activas, caso seja necessário.
  - Passo 11. Negociação dos contratos de alto nível com o cliente.
  - Passo 12. A execução prossegue com a verificação periódica dos contratos, afinação dos parâmetros das políticas activas, actualização dos equipamentos da rede e renegociação dos contratos, consoante necessário.

As principais diferenças relativamente ao trabalho de [Lewis 99a] são: existem vários níveis de abstracção e não apenas serviços; e as políticas activas irem além da simples monitorização de níveis de serviço e produção de relatórios, ao incluírem controlo e planeamento.

A execução de alguns passos da metodologia atrás apresentada corresponde mais a uma arte do que a uma ciência, pois não é possível saber com precisão qual a contribuição de cada parâmetro do sistema para a satisfação dos requisitos de alto nível. O passo mais complexo da metodologia é o passo 4, onde se pretende relacionar os parâmetros que podem ser medidos e controlados com os parâmetros de QoS

pretendidos. Na realidade, pode ser necessário retornar a este passo várias vezes quando, no passo 10, se colocarem as políticas activas a funcionar no sistema real, sendo este um processo heurístico.

Tome-se como exemplo o caso de um serviço via *web*, e um requisito de alto nível descrito como a rapidez de execução do serviço. O tempo que demora a executar o serviço pode ser medido com precisão na aplicação do utilizador, mas o número de factores de que depende é grande e difícil de determinar. A rapidez depende do número e complexidade das páginas carregadas, da capacidade de processamento dos computadores envolvidos, da existência de *caches*, do débito, atraso e taxa de erros das redes envolvidas, entre outros factores. Assim, neste exemplo, seria possível medir o valor da QoS de alto nível, mas modificar a QoS obtida pode obrigar a redesenhar todos os sistemas envolvidos. Noutros casos pode nem ser fácil medir a QoS de alto nível. Por exemplo, como medir a satisfação dos utilizadores? Embora se possa ter alguns indicadores como a rapidez, disponibilidade e fiabilidade do serviço, pode só ser possível ter uma ideia mais precisa recorrendo a inquéritos aos utilizadores [Cochrane 91].

Este problema é especialmente difícil de resolver pois é necessário fazer convergir o refinamento dos requisitos de alto nível nos parâmetros disponíveis nos objectos geridos. Embora em muitos casos se tenha de recorrer à arte e intuição dos gestores da rede e sistemas, há alguns métodos científicos que podem ajudar nesta tarefa.

### **3.5.1 Comparação de Sistemas Periciais**

O domínio da Inteligência Artificial propõe vários tipos de sistemas periciais que permitem simular o trabalho de um especialista humano. Em [Gardner 96] [Meira 97] [Lin 98] e [Lewis 99c] são sugeridos e comparados vários tipos de sistemas periciais que permitem correlacionar eventos para detectar situações com problemas, e resolvê-las. No resto desta subsecção apresentam-se resumidamente alguns deles.

A lógica difusa [Kruse 94] permite traduzir valores numéricos de propriedades em probabilidade de pertença a conjuntos difusos. A figura 3.4 ilustra como se pode converter o valor da carga em classificações difusas como “baixa”, “média” e “alta”. Depois trabalha-se com regras em termos das classificações difusas. Por exemplo: se a

carga é “alta” e o débito é “baixo”, então a qualidade é “fraca”, o aviso é “alerta”, e o redirecionamento de pedidos é “pequeno aumento”. Finalmente, depois de aplicar as regras difusas, deve-se converter os termos difusos obtidos de volta para parâmetros numéricos que possam ser utilizados no sistema real, invertendo-se o passo inicial. Uma vantagem da lógica difusa é poder trabalhar-se com conceitos imprecisos, vagos, ou incertos. No entanto, é necessário um sistema adicional para definir as regras a utilizar. [Valimaa 95] apresenta um exemplo de um sistema de geração de parâmetros de QoS utilizando lógica difusa.

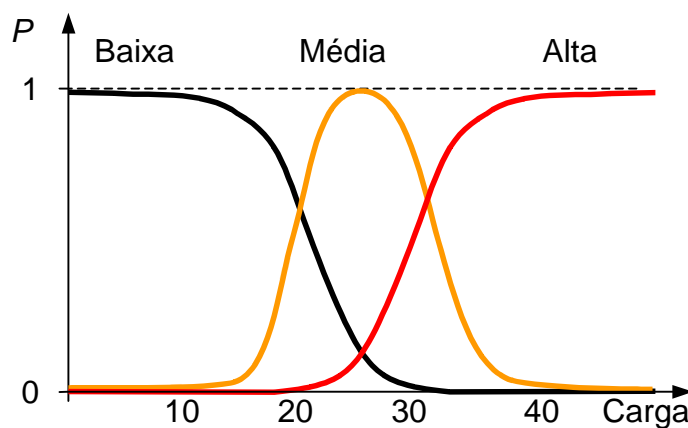


Figura 3.4: Conceitos difusos de carga “baixa”, “média” e “alta”.

As redes neuronais [Lippmann 87] simulam o funcionamento dos neurónios humanos, podendo ser utilizadas para aprender quais as respostas pretendidas para estímulos dados. A figura 3.5 ilustra o funcionamento de um neurónio artificial. As entradas são multiplicadas por pesos e somadas, obtendo-se uma média ponderada que é passada por uma função, normalmente não linear, para produzir a saída. Os pesos das entradas podem ser ajustados por um algoritmo de aprendizagem, aplicando sequências de treino com respostas conhecidas. As redes neuronais normalmente têm várias camadas com múltiplos neurónios, o que permite uma aprendizagem de padrões mais complexos do que com apenas uma única camada. Uma desvantagem das redes neuronais é a necessidade de realizar um treino prévio com grandes quantidades de pares de entradas e saídas, que normalmente não estão disponíveis em gestão. Além disso, as redes neuronais são incapazes de explicar o porquê das respostas produzidas. [Vieira 96] apresenta um exemplo de um sistema baseado em redes neuronais para detecção e classificação de falhas num ambiente de gestão de sistemas Unix. [Ensel 99]

utiliza redes neurais para correlacionar tráfego em várias máquinas e assim inferir dependências entre serviços nas várias máquinas.

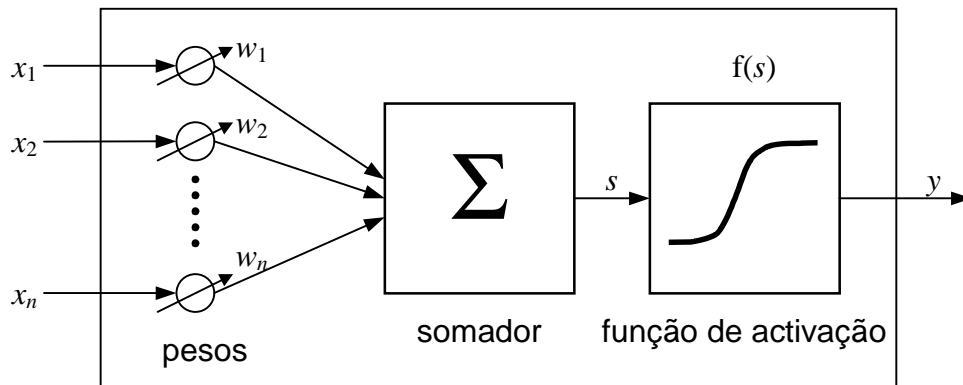


Figura 3.5: Neurónio artificial de uma rede neuronal.

Os sistemas de raciocínio baseados em regras (*rule-based reasoning*) têm uma máquina de inferência, que utiliza um conjunto de regras de inferência para, com base nos factos detectados, decidir as acções a tomar. Uma grande desvantagem destes sistemas é ter de se prever regras para todos os casos, o que é complicado de fazer e manter actualizado.

Os sistemas de raciocínio baseados em casos (*case-based reasoning*) contêm um registo de casos anteriores e as respectivas soluções, com os quais comparam o caso actual, para obter a melhor solução. Estes sistemas guardam no registo a experiência ganha para a solução dos novos casos, mas são incapazes de encontrar uma solução se não encontrarem nenhum problema semelhante no seu registo.

Os sistemas de raciocínio baseados em modelos (*model-based reasoning*) utilizam modelos da estrutura e comportamento dos sistemas reais para detectar os problemas por comparação entre a observação e a previsão dos modelos. Estes sistemas periciais têm a vantagem de poder detectar problemas que nunca ocorreram, mas é muito complicado de capturar nos modelos toda a estrutura e comportamento dos sistemas modelados.

Os sistemas baseados em autómatos, ou grafos de transição de estados (*state-transition graph*), utilizam um grafo de arcos e estados, que é percorrido até se encontrar uma solução. A cada estado está associada uma acção a realizar, e, consoante o seu resultado, segue-se por um arco diferente para um novo estado até resolver o

problema. Embora os grafos possam ser reutilizados para diferentes tipos de componentes, exigem uma cuidadosa construção, e não têm capacidade de aprendizagem, ou adaptação a novas situações.

Os sistemas baseados em correlação por codificação (*coding approach*), também chamados livro de código (*codebook*), utilizam uma matriz onde cada linha corresponde a um evento, e cada coluna a um problema. A correlação é feita através da escolha nesta matriz do problema cuja codificação mais se aproxima ao padrão de eventos detectado. Este método é computacionalmente muito eficiente, mas é complicado construir a matriz inicial, e é necessário recompilá-la por cada modificação. [Yemini 96] usa este método para detecção de falhas.

Os sistemas baseados em redes bayesianas, também chamadas redes causais, ou redes de crença, utilizam um grafo acíclico dirigido, no qual os nós representam conhecimento incerto, e os arcos dependências de causas e efeitos entre os nós. Os arcos expressam a distribuição de probabilidade para cada nó efeito, com base nas probabilidades dos nós causa. Partindo de um conjunto de evidências, isto é, nós com valores conhecidos, é possível calcular as probabilidades condicionais dos outros nós. Este método permite trabalhar com conhecimento incompleto, mas apresenta as mesmas desvantagens dos sistemas baseados em regras. [Hood 98] usa redes bayesianas para detecção de falhas.

É possível construir modelos ARMA (*AutoRegressive Moving Average*) de sistemas a partir de históricos de medições, podendo depois estudar-se a aplicação de controladores aos sistemas. [Parekh 01] aplica este método para construir um modelo de um servidor de correio electrónico. Depois constrói um controlador que permite controlar a carga do servidor de forma a oferecer uma certa QoS.

Da comparação dos vários sistemas, conclui-se que não há nenhum que seja óptimo para resolver os diferentes tipos de problemas de gestão de redes e sistemas. Os factores mais importantes na escolha de um sistema são: facilidade de construção de um modelo da rede e dos sistemas; facilidade de aprendizagem, ou adaptação a novas situações; exactidão das respostas fornecidas; facilidade de implementação; e desempenho do sistema.

Todos os sistemas são úteis na detecção de falhas, pelo que também podem dar boas indicações sobre a QoS que se está a obter. No que diz respeito a agir no sentido de

melhorar, ou assegurar a QoS, qualquer deles necessita que um especialista no problema em causa programe a solução.

### **3.5.2 Funcionamento Interno das Políticas Activas**

Nenhum dos tipos de sistemas periciais descritos na subsecção anterior (3.5.1) prevê uma arquitectura com vários níveis de abstracção, ou como os requisitos de gestão podem ser refinados até chegar a parâmetros dos sistemas geridos. Desta forma, numa arquitectura como a que se propõe nesta tese, os sistemas periciais poderiam servir apenas para construir parte do comportamento interno das políticas activas. No entanto, nesta tese propõe-se a utilização de um sistema mais simples, adaptado à gestão de qualidade de serviço, e que permite cobrir os casos de utilização mais frequente. Porém, não se exclui a utilização individual de técnicas da subsecção anterior.

Tal como descrito anteriormente na secção 3.3 e ilustrado na figura 3.3, as políticas activas começam por monitorizar os objectos geridos sob sua responsabilidade e outras políticas activas em níveis de abstracção inferiores. A figura 3.6 detalha o modo como funciona o bloco de monitorização de uma política activa e como são calculados os seus parâmetros de QoS. O primeiro passo é amostrar os parâmetros do sistema relevantes à determinação da QoS. Estes parâmetros podem ser parâmetros de QoS de outras políticas activas, eventos, ou valores lidos da MIB de objectos geridos. O segundo passo é realizar conversões de formato da informação lida. O terceiro passo é realizar um pré-processamento dos dados, por exemplo para filtrar picos. Finalmente, o último passo é calcular os parâmetros de QoS da própria política activa.

A amostragem dos dados é parametrizada pela frequência de amostragem que deve ser suficientemente grande para não se perder informação relevante. O teorema da amostragem, devido a Nyquist, diz que, para ser possível reconstruir o sinal a partir das suas amostras, a frequência de amostragem deve ser pelo menos o dobro da maior frequência existente no sinal a amostrar. Embora para minimizar o processamento necessário, e os custos correspondentes, se deva escolher uma frequência de amostragem o mais baixa possível, é normal utilizar uma frequência de amostragem 10 vezes superior à mínima [Franklin 87b], de forma a reduzir o atraso na resposta. [Yoshihara 99] propõe um método de ajuste dinâmico da frequência de amostragem que leva em conta tanto o consumo de largura de banda necessário às amostragens, como os

requisitos das tarefas de gestão. O grupo de medição de desempenho do IETF propõe [RFC 2330] um mecanismo de amostragem com intervalos aleatórios, como forma de evitar possíveis efeitos de sincronização causados pelo tráfego de amostragem, e ainda para evitar que se possa prever os instantes das amostragens e assim manipular as medições. Uma vez que se pretende que as políticas activas estejam junto, ou mesmo embebidas, nos objectos geridos, o tráfego necessário às amostragens será nulo, ou pouco significativo, pelo que se achou suficiente a utilização de uma frequência de amostragem fixa.

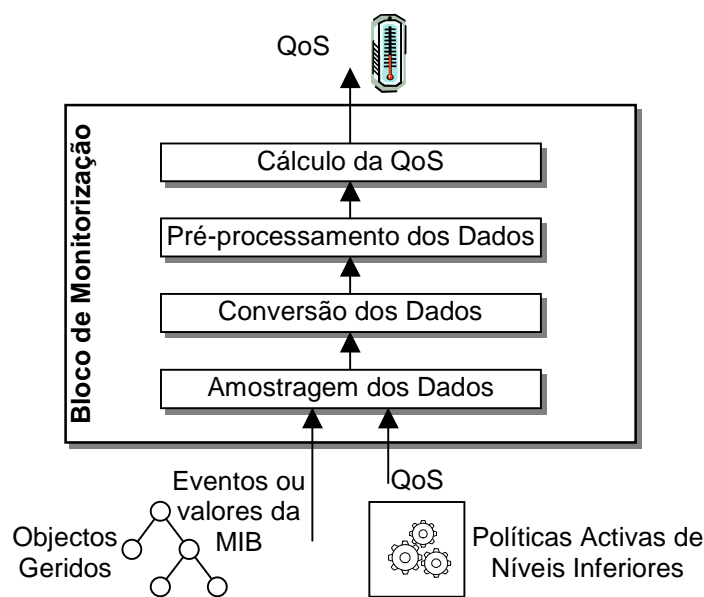


Figura 3.6: Bloco de monitorização de uma política activa.

Os parâmetros do sistema podem ser classificados consoante o seu domínio, podendo ser:

- **Discretos**, quando apenas alguns valores são permitidos, por exemplo: ligado/desligado, alto/baixo/parado. A estes valores discretos devem estar associados números inteiros.
- **Contínuos**, quando qualquer valor real é permitido.

Os parâmetros do sistema podem ainda ser classificados consoante possam, ou não, ser modificados, podendo ser:

- **Apenas de leitura**, quando não podem ser modificados.
- **Leitura e escrita**, quando podem ser modificados dentro de uma determinada gama.



A conversão do formato dos dados pode consistir em vários tipos de operações, destacando-se os conversores implementados:

- Calcular a diferença entre amostras consecutivas. Esta conversão serve para, por exemplo, converter o valor de um contador do total de pacotes perdidos no número de pacotes perdidos no intervalo de amostragem. Uma segunda aplicação desta conversão permite obter a derivada do indicador, ou seja, uma indicação se houve mais, ou menos pacotes perdidos no intervalo.
- Acumular ao valor anterior. Esta conversão, inversa da anterior, permite obter o total acumulado a partir do valor obtido em cada intervalo de amostragem.

O pré-processamento dos dados permite resumir os dados ao longo do tempo, ou sobre um conjunto de objectos geridos. Destacam-se os seguintes tipos de operações implementadas:

- Obter a média. No caso de uma média no tempo, pode usar-se uma média móvel de peso exponencial (EWMA), que dá um peso menor aos dados mais antigos; uma média móvel de peso constante (UWMA), que dá peso igual a todos os dados dentro de uma janela deslizante de dimensão fixa; ou uma média dentro de um intervalo fixo. No caso de uma média sobre os dados recolhidos de um conjunto de objectos geridos, também se pode ter pesos iguais para todos, ou não. Por exemplo, pode ser calculada a média dos atrasos sofridos pelos pacotes em cada objecto gerido pesada pelo número de pacotes em cada um, ou com pesos iguais para todos.
- Obter o quantil de ordem  $p$ , definido como o valor para o qual a função densidade de probabilidade passa de um valor menor ou igual a  $p$ , para um valor maior que  $p$ .
- Obter o máximo, ou o mínimo.
- Obter a variância, o desvio padrão, ou outro momento.

Por fim, o bloco de monitorização termina com o cálculo dos parâmetros de QoS com base nos últimos valores conhecidos dos dados pré-processados, utilizando uma função de mapeamento dos parâmetros do sistema para a QoS do serviço, que no caso geral será:

$$QoS(serviço) = f(P_1, P_2, \dots, P_n) \quad (3.1)$$

Nos problemas mais simples, a função de mapeamento dos parâmetros do sistema para a QoS é linear, podendo a equação ser rescrita como:

$$QoS(serviço) = \sum_{i=1}^n a_i \cdot P_i \quad (3.2)$$

onde os  $P_i$  são os dados pré-processados correspondentes a cada parâmetro do sistema, e os  $a_i$  são os pesos de cada parâmetro do sistema, de acordo com o que se pensa que é a importância de cada um para o parâmetro de QoS em causa e o seu valor normal.

Para o caso de contratos do tipo relativo, a forma geral da função de mapeamento será do tipo:

$$QoS(serviço) = \frac{P_{abs}}{a \cdot P_{ref}} \quad (3.3)$$

onde  $P_{abs}$  corresponde a uma indicação absoluta da QoS,  $P_{ref}$  corresponde ao parâmetro de referência, e a constante  $a$  permite ajustar a referência para o valor pretendido, de forma a que o valor normal da  $QoS(serviço)$  seja 1.

Como exemplo de um caso em que estas formas gerais não se aplicam, tem-se um serviço que depende de múltiplos objectos, onde a disponibilidade do serviço depende da disponibilidade simultânea de todos os objectos que o compõem:

$$QoS(Disponibilidade) = \prod_{i=1}^n Disponibilidade_i \quad (3.4)$$

e a probabilidade do serviço resultar num erro depende da probabilidade de todos os objectos que o compõem funcionarem correctamente em simultâneo:

$$QoS(Perro) = 1 - \left( \prod_{i=1}^n (1 - Perro_i) \right) \quad (3.5)$$

O bloco de gestão reactiva é o bloco que se encontra após o bloco de monitorização nas políticas activas. O bloco de gestão reactiva dá uma resposta rápida a certo tipo de situações. A figura 3.7 mostra o funcionamento interno do bloco de gestão reactiva. As regras de actuação definem quais as acções a executar de forma a fazer respeitar o contrato, descrevendo-se na próxima secção (3.6) as regras de actuação predefinidas. Nas situações mais habituais, estas regras de actuação comparam a QoS com determinados limiares, detectando situações que desencadeiam a execução de acções de gestão, ou o envio de eventos. Os limitadores permitem verificar se as acções a executar fazem sentido, limitando as variáveis modificadas às gamas permitidas. A geração de excepções verifica se o contrato da política activa foi violado, tomando as

acções de excepção previstas. Também é possível gerar excepções nos casos em que os limitadores actuaram, significando que a política activa já não está a conseguir actuar no bom sentido, pelo que será de esperar que a qualidade de serviço continue a degradar-se.

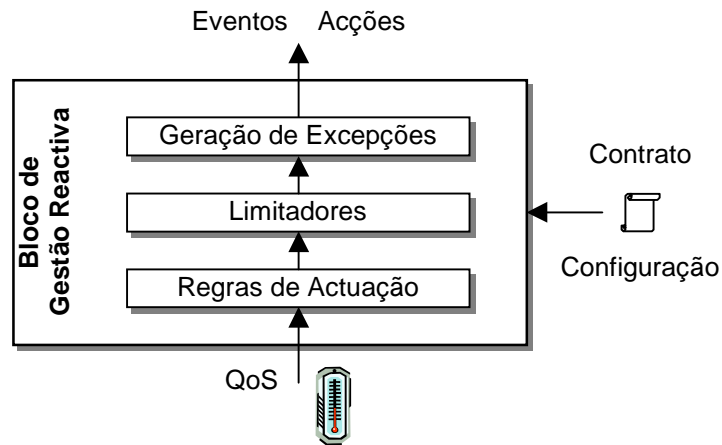


Figura 3.7: Bloco de gestão reactiva de uma política activa.

Finalmente, o último bloco que compõe uma política activa é o bloco de gestão proactiva, que efectua tarefas de planeamento. A figura 3.8 mostra o funcionamento interno do bloco de gestão proactiva. A história guarda a evolução dos parâmetros de QoS dentro de determinadas janelas temporais. As regras de actuação definem como são modificados os parâmetros internos da política activa em resultado do planeamento. Desta forma, o bloco de planeamento, em vez de actuar directamente sobre os objectos geridos, modifica a configuração da própria política activa, por exemplo ajustando os limiares de actuação do bloco de gestão reactiva, consoante a análise da evolução da história de determinado parâmetro. Esta retroactividade, se for bem utilizada, permite assegurar os contratos com maior precisão. Na realidade, mesmo que os limiares de actuação escolhidos inicialmente não sejam os melhores, é possível corrigi-los em tempo real com o bloco de gestão proactiva. A maior parte das regras de actuação pode ser utilizada quer no bloco de gestão reactiva, quer no bloco de gestão proactiva, havendo algumas regras específicas para tarefas de planeamento, como por exemplo: análise dos desvios observados, ou análise de tendências. Os limitadores, mais uma vez, permitem verificar se os novos valores dos parâmetros fazem sentido, limitando-os às gamas permitidas, antes de as modificações serem realizadas. Por fim, a geração de excepções verifica se os limitadores actuaram, o que significa que já não será possível

reconfigurar a política activa para atingir os objectivos pretendidos. Os eventos resultantes destas excepções são, por isso, de maior gravidade que os gerados pelo bloco de gestão reactiva.

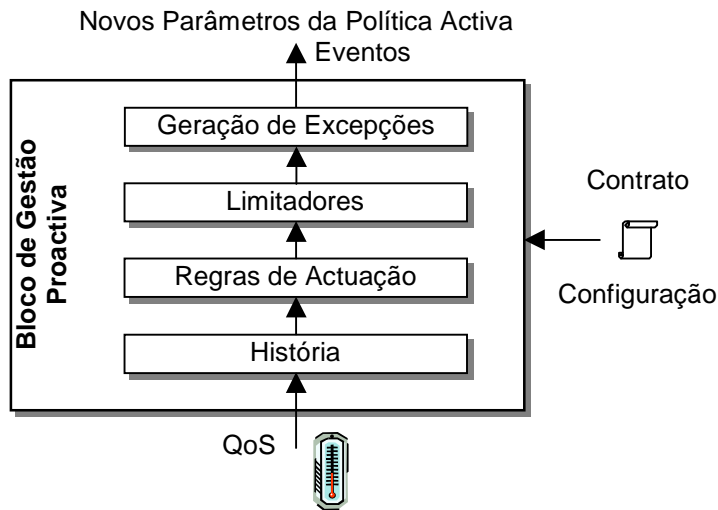


Figura 3.8: Bloco de gestão proactiva de uma política activa.

A estrutura dos blocos apresentados nesta secção tem grandes semelhanças com a estrutura da SMF de atributos e objectos de medida apresentada na secção 2.2.3.1. A diferença principal é que aquela SMF é genérica, enquanto nesta secção concretizaram-se operações de monitorização, controlo e planeamento com o objectivo de calcular a QoS e agir no sentido de a assegurar.

### 3.5.3 Construção do Comportamento das Políticas Activas

Sendo os parâmetros de QoS de uma política activa definidos em termos de um certo conjunto de parâmetros do sistema gerido, conforme expresso nas equações 3.1 a 3.3, importa agora pensar como deve ser construído o comportamento das políticas activas, ou seja, as regras de actuação e a sua parametrização, para assegurar o contrato feito sobre os parâmetros de QoS.

Para saber como assegurar o valor de um dado parâmetro de QoS, importa escolher uma função de mapeamento desse parâmetro de QoS em parâmetros do sistema que possam ser modificados. Veja-se o caso, referido no início desta secção, da rapidez de execução de um serviço via HTTP. Se estiver disponível o tempo de cada pedido HTTP, o parâmetro de QoS pode ser calculado muito simplesmente com uma função de

mapeamento com este único parâmetro do sistema. No entanto, como este parâmetro apenas pode ser monitorizado, sem poder ser modificado, terá de se construir uma outra função de mapeamento que inclua um número máximo de parâmetros do sistema que possam ser modificados. Esta função de mapeamento pode incluir outros parâmetros do sistema, possivelmente de níveis de abstracção inferiores, mesmo que a precisão na obtenção do valor da QoS seja menor. Neste exemplo, poderia ser utilizado o número e complexidade das páginas carregadas, a dimensão de *caches*, o débito, atraso e taxa de erros das redes envolvidas, a capacidade de processamento dos computadores envolvidos, etc.

A figura 3.9 ilustra um exemplo de um parâmetro de QoS que depende de três parâmetros do sistema. Além disso, mostra a gama de valores permitidos para a QoS com um contrato típico que especifica um valor óptimo e um desvio máximo para a QoS.

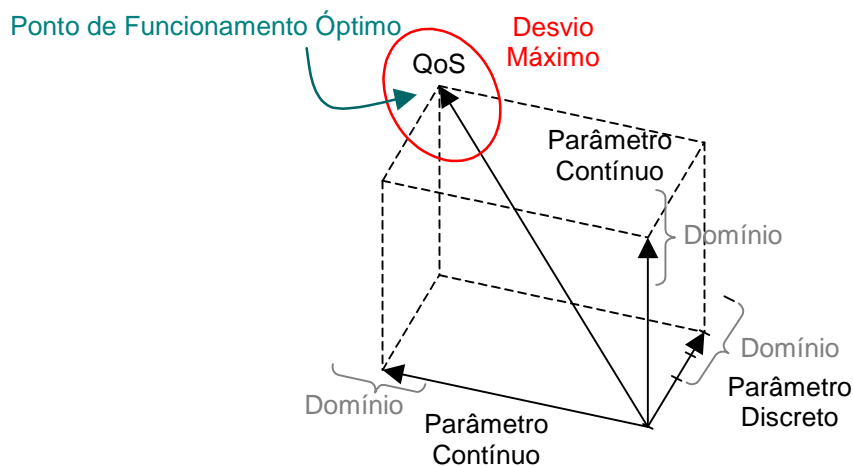


Figura 3.9: Modelo de um parâmetro de QoS de gestão.

Para assegurar o contrato, a política activa deve reagir, modificando os parâmetros do sistema que influem na QoS, de modo a manter a QoS dentro da gama aceitável. Se nenhum dos parâmetros do sistema puder ser modificado, então a política activa limita-se a realizar tarefas de monitorização, calculando a QoS obtida, e a gerar eventos. Se houver parâmetros que possam ser modificados, então a política activa utiliza as regras de actuação para os modificar à medida do necessário.

O conjunto das regras de actuação deve implementar um algoritmo de ajuste dos parâmetros sob controlo da política activa. Este algoritmo deve procurar modificar os

parâmetros do sistema no sentido de aumentar a QoS quando esta está baixa, de forma a evitar violações do contrato. Além disso, também deve modificar os parâmetros no sentido de baixar a QoS quando esta está alta, de forma a consumir o mínimo de recursos. Caso não seja possível assegurar o contrato, a política activa deve realizar as acções de excepção que o contrato especifica. A figura 3.10 descreve este algoritmo genérico.

```
Iniciar o serviço S com parâmetros de controlo P
Enquanto o serviço S estiver activo
  Monitorizar a qualidade de serviço Q
  calculada com a função de mapeamento F
  Se Q é melhor que o acordo A então
    modificar ligeiramente P no sentido de
    economizar recursos (piorando Q)
  Senão Se Q é pior que o acordo A então
    modificar P no sentido de melhorar Q
  Se P atingiu o limite então
    Gerar excepção associada ao acordo A
```

Figura 3.10: Algoritmo genérico de ajuste de parâmetros.

As modificações nos parâmetros devem ser determinadas de acordo com o modelo do sistema, levando em conta o domínio de cada parâmetro e o peso com que cada um contribui para a QoS. Além disso, as modificações nos parâmetros devem ser sempre menores quando se actua no sentido de piorar a QoS, do que quando se actua no sentido de melhorar a QoS, de forma a minimizar a probabilidade de violação do contrato. É possível obter uma convergência mais rápida para os objectivos se se modificar mais do que um parâmetro de controlo de cada vez. No entanto, modificações grandes demais podem conduzir a oscilações indesejáveis em torno do ponto de funcionamento óptimo, havendo assim um compromisso entre rapidez e suavidade de convergência.

Na subsecção seguinte apresentam-se alguns critérios que permitem avaliar a eficácia de um sistema de controlo.

### 3.5.4 Critérios de Avaliação

Com o objectivo de avaliar os sistemas de gestão baseados em políticas activas, serão utilizados os seguintes critérios, baseados nos critérios de avaliação de um sistema de controlo propostos em [Chiu 89]:

**Eficiência:** a eficiência de utilização de um recurso é definida pela aproximação ao valor ideal, ou ao objectivo. Tanto a sobreutilização, ou sobrecarga, como a subutilização são igualmente indesejadas.

**Justiça:** o critério *maxmin* de justiça é habitualmente aplicado quando múltiplos utilizadores partilham múltiplos recursos. Para utilizar este critério, os utilizadores são divididos em classes de equivalência de acordo com o recurso mais limitativo de cada um. O critério *maxmin* pretende que os utilizadores da mesma classe devem ter uma fracção igual desse recurso.

**Distribuição:** o método de controlo deve ser distribuído. Um esquema centralizado necessita de ter completo conhecimento do estado do sistema, necessitando de transferir grandes quantidades de informação. Pretende-se que o sistema funcione com um mínimo de informação transportada na rede. Este assunto já foi analisado com maior profundidade na secção 2.3.2.

**Convergência:** requer-se que o sistema de controlo convirja. A convergência é habitualmente medida pela rapidez com que o sistema se aproxima do objectivo, partindo de qualquer estado, e pela amplitude das oscilações em torno do ponto óptimo. O tempo necessário até chegar ao objectivo determina a rapidez (*responsiveness*), e a dimensão das oscilações determina a suavidade (*smoothness*) do controlo, tal como ilustrado na figura 3.11. Idealmente, ambos devem ser pequenos.

Para garantir a convergência, é necessário que as actuações sobre o sistema sejam na direcção correcta. Isto é, quando se deve actuar no sentido de diminuir um certo parâmetro, deve-se garantir que esse parâmetro não aumenta, e quando se deve actuar no sentido de aumentar um certo parâmetro, deve-se garantir que esse parâmetro não diminui. Este é o princípio de realimentação negativa.

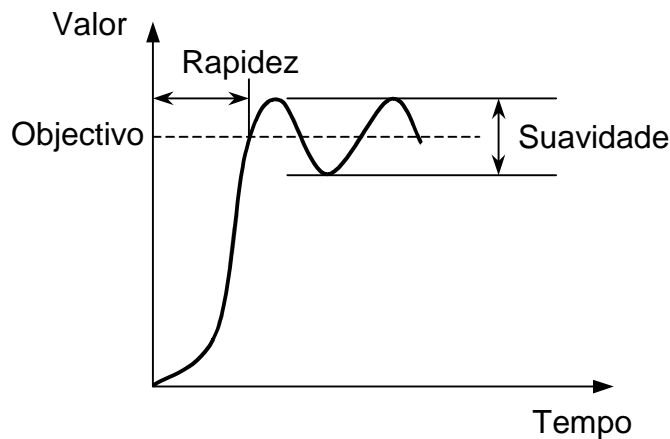


Figura 3.11: Avaliação da convergência para um objectivo.

### 3.6 REGRAS DE ACTUAÇÃO PREDEFINIDAS

A maior parte das políticas activas pode ser construída compondo e parametrizando devidamente regras de actuação predefinidas. Nesta secção apresentam-se alguns exemplos de regras predefinidas.

A regra *threshold* actua com base em dois limiares e histerese conforme ilustrado na figura 3.12. Quando a QoS é baixa, e sobe, ultrapassando o limiar de subida, a acção de subida é executada, e fica-se no estado em que a QoS é alta. De forma semelhante, quando a QoS é alta, e desce abaixo do limiar de descida, a acção de descida é executada, ficando-se no estado em que a QoS é baixa. Uma vez que, nas situações normais, se pretende que a QoS não atinja um determinado valor limite, os limiares de subida e descida são expressos como uma percentagem desse valor limite da QoS. Esta regra serve de base a várias outras regras e variantes.

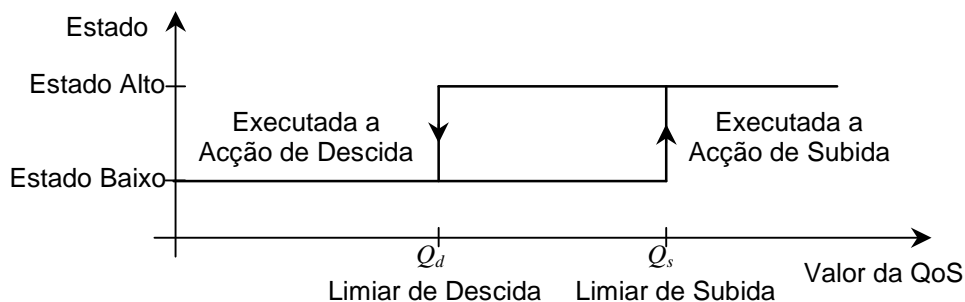


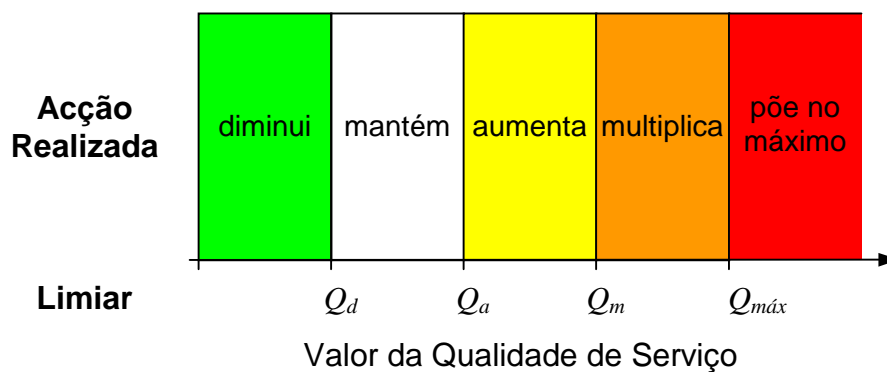
Figura 3.12: Funcionamento da regra básica de limiares com histerese.



Uma outra regra de actuação semelhante a esta é a regra `threshold_increment` que mantém os dois limiares de funcionamento, mas incrementa um determinado parâmetro de controlo quando se está acima do limiar superior,  $Q_s$ , e decrementa esse parâmetro de controlo quando se está abaixo do limiar inferior,  $Q_d$ . O valor do incremento, ou decremento, ao parâmetro de controlo é configurável, sendo feito uma vez por cada amostra tirada, enquanto os limiares forem excedidos. Uma variante desta regra é a regra `threshold_decrement`, que tem o incremento negativo. Estas regras servem para as situações em que o parâmetro de controlo deve ser aumentado, ou diminuído, para melhorar a QoS. Normalmente são combinadas com limitadores para que o parâmetro de controlo se mantenha dentro da gama de valores válidos.

A regra `increase` é semelhante a esta, aumentando um determinado parâmetro de controlo quando se está acima do limiar superior,  $Q_s$ , mas decrementando esse parâmetro de controlo quando se está abaixo do limiar inferior,  $Q_d$ , até ao valor pretendido para o parâmetro de controlo. O valor do incremento, ou decremento, ao parâmetro de controlo é configurável, sendo feito uma vez por cada amostra tirada, enquanto os limiares forem excedidos. Uma variante desta regra é a regra `decrease`, que funciona com o parâmetro de controlo abaixo do valor pretendido, decrementando-o sempre que se excede o limiar superior. Estas regras servem para as situações em que o parâmetro de controlo tem um valor normal, que pode ser temporariamente aumentado, ou diminuído, para melhorar a QoS.

A regra `adjust_increment` inclui mais limiares sobre o valor da QoS, ajustando o incremento consoante a zona de funcionamento, conforme ilustrado na figura 3.13. As primeiras três zonas têm funcionamento semelhante à regra `threshold_increment`. Mas, se a QoS for superior a  $Q_m$ , então o incremento passa a ser multiplicativo, e se a QoS for superior a  $Q_{máx}$ , então o parâmetro de controlo é colocado no valor máximo possível. Também é possível ter várias variantes desta regra, com limiares inferiores, ou simultaneamente inferiores e superiores, para incrementos multiplicativos e colocação no máximo. Estas regras servem para as mesmas situações que as regras `increase` e `decrease`, mas permitem que o desvio do parâmetro de controlo seja muito mais rápido quando a QoS se aproxima dos limites impostos, do que a voltar ao normal, de acordo com o algoritmo genérico de ajuste dos parâmetros descrito anteriormente na figura 3.10.

Figura 3.13: Funcionamento da regra `adjust_increment`.

A regra `on/off` modifica o valor de um determinado parâmetro de controlo de uma certa percentagem (modificação multiplicativa), repondo o valor original quando o limiar inferior é ultrapassado. A regra pode não voltar a disparar até o limiar oposto ser cruzado, ou pode ser reiniciada automaticamente passado um certo tempo. Uma variante desta regra é ter a modificação aditiva em vez de multiplicativa. Estas regras permitem modificar um parâmetro de controlo num dado sentido quando a QoS piora, repondo imediatamente a situação normal quando a QoS volta a ser aceitável.

A regra `count` conta o número de vezes seguidas que uma certa condição é verdadeira, incrementando um determinado parâmetro de controlo sempre que a contagem máxima for atingida. Além disso, também conta o número de vezes seguidas que a condição é falsa, decrementando esse parâmetro de controlo sempre que a contagem máxima for atingida. A contagem máxima pode ser diferente para cada sentido. Esta regra permite que se actue apenas no caso de situações repetidas. Por exemplo, será possível modificar um parâmetro de controlo sempre que, num dado período, haja descarte de pacotes e, no sentido inverso, apenas quando houver 10 períodos seguidos sem descarte de pacotes.

A regra `time` permite gerar valores com base apenas na passagem de tempo.

Como exemplo de uma regra de planeamento, a regra `pid_control` realiza um controlador PID. Esta regra calcula um novo valor para um parâmetro de controlo com base nos valores actuais e anteriores da QoS, pela soma de um termo proporcional, integrativo e derivativo, de acordo com o exposto no apêndice B. Esta regra permite, por exemplo, utilizando uma malha de realimentação, conforme ilustrado na figura 3.14, servir para ajustar um parâmetro de um objecto gerido, ou de outra regra de actuação, de

forma a melhor atingir um objectivo de QoS fixado num SLA. Consegue-se, assim, minorar a dificuldade em fixar os limiares de actuação nas regras precedentes.

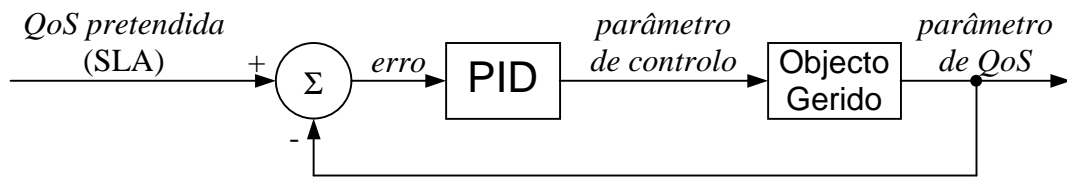


Figura 3.14: Exemplo de utilização da regra `pid_control`.

As regras de actuação apresentadas nesta secção são regras genéricas que se podem aplicar a qualquer tipo de problemas, desde que seja possível prever, de alguma forma, os efeitos de modificar os parâmetros de controlo. Normalmente as políticas activas podem ser construídas compondo e parametrizando várias destas regras com os blocos de amostragem, conversão, pré-processamento, e limitadores adequados, de forma a ajustar apropriadamente os parâmetros de controlo dentro das gamas de valores normais. Consegue-se, assim, realizar na prática o algoritmo genérico de ajuste dos parâmetros dos sistemas geridos descrito anteriormente na figura 3.10.

Para problemas mais específicos, ou para otimizar as soluções de gestão, pode ser necessário desenvolver novas regras, ou políticas activas especialmente adaptadas a cada problema particular. Para isso pode ser necessário utilizar linguagens que permitam aceder directamente aos protocolos de gestão em uso, tais como as linguagens apresentadas anteriormente na secção 2.9.

Nos capítulos 4 e 5 apresentam-se cenários completos onde são utilizadas estas regras predefinidas e também algumas regras específicas.

### 3.7 ARQUITECTURA DE GESTÃO

A aplicação para realizar a gestão da rede, sistemas, serviços e processos de negócio corresponde à hierarquia de políticas activas que é utilizada para desempenhar as tarefas de gestão e assegurar os contratos. Os níveis da hierarquia correspondem aos níveis de abstracção. No entanto, esta hierarquia é conceptual, o que levanta um problema: como colocar as políticas activas na rede a gerir os sistemas?

Propõe-se a utilização de uma arquitectura distribuída, tal como descrito na secção 2.3, com gestão por delegação. Pretende-se, com uma arquitectura deste tipo, colocar as políticas activas junto dos objectos geridos com o objectivo de manter um controlo apertado sobre a QoS, e, simultaneamente, minimizar o tráfego de gestão, aumentando a escalabilidade e a tolerância a falhas do sistema de gestão. Por outro lado, a utilização de gestão por delegação torna o sistema mais flexível, permitindo realizar modificações durante o funcionamento do sistema.

A arquitectura de gestão está esquematizada na figura 3.15. No topo estão as estações de gestão, com os gestores de topo, que implementam as políticas activas dos níveis de abstracção mais altos. Aqui pretende-se ter uma visão global do estado da rede e da QoS que está a ser obtida. Os gestores intermédios implementam as políticas activas com âmbito de rede local, devendo existir em número adequado à carga das tarefas de gestão e à topologia da rede. Finalmente, há gestores nos equipamentos, ou a eles associados, onde são colocadas as políticas activas específicas aos mesmos.

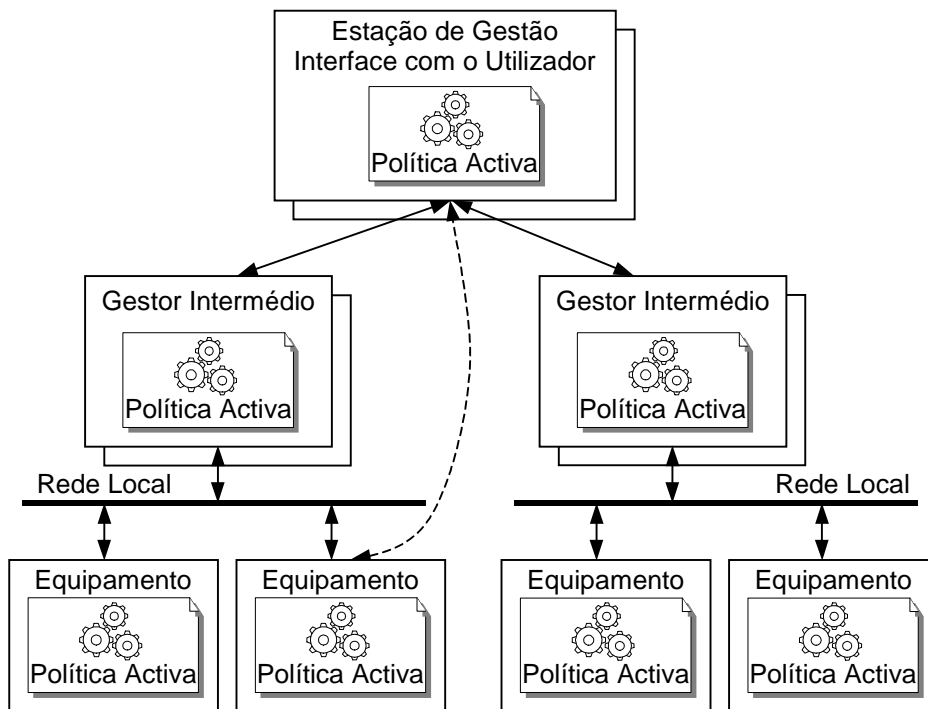


Figura 3.15: Arquitectura para implementar a hierarquia de políticas activas.

Pensa-se assim, ter conseguido obter uma arquitectura flexível, que se pode adaptar dinamicamente, tanto a diferentes topologias de rede, como a diferentes necessidades de processamento de tarefas de gestão.

### **3.8 BENEFÍCIOS DAS POLÍTICAS ACTIVAS**

Há cinco aspectos onde a aproximação por políticas activas traz vantagens. O primeiro é a transparência no que diz respeito a certos aspectos da operação dos sistemas, tendo-se a possibilidade de apenas monitorizar a QoS que se está a obter no nível de abstracção adequado. Cada política activa faz o seu melhor para cumprir o seu contrato, e desde que o contrato não seja violado, as outras entidades não precisam de saber como isso é conseguido. A existência de vários níveis de abstracção permite ainda que, quando há um problema num dos níveis de abstracção mais baixos, que se consiga resolver, este fique encapsulado nesse nível de abstracção, não afectando os níveis de abstracção superiores. Além disso, os diferentes níveis de abstracção permitem a um gestor trabalhar no grau de abstracção mais alto possível, escondendo os detalhes dos níveis inferiores.

A segunda vantagem é a possibilidade de assegurar SLAs. Para além de se permitir avaliar a QoS obtida nos serviços e processos de negócio, vai-se mais além, ao actuar sobre os sistemas geridos, no sentido de garantir os requisitos de gestão especificados nos SLAs.

A terceira vantagem é a simplicidade do ambiente. A maior parte das políticas activas pode ser construída com base nas regras de actuação predefinidas, bastando parametrizá-las de forma adequada. Além disso, as políticas activas podem também ser reutilizadas para construir novas aplicações de gestão, consoante os objectivos do gestor.

A quarta vantagem é a perspectiva proporcionada por um ambiente de políticas activas. Normalizando um ambiente de políticas activas, os fabricantes têm regras simples que lhes permitem incorporar funcionalidades próprias para problemas específicos. Assim, os fabricantes são estimulados a oferecer soluções de gestão sob a forma de políticas activas orientadas para a procura de soluções para problemas, com o

objectivo de assegurar o funcionamento de serviços, e não tanto para oferecer ferramentas gerais de gestão.

Finalmente, as redes e os sistemas estão a tornar-se ambientes distribuídos. Os aspectos de processamento distribuído devem ser tomados em consideração à medida que o sistema gerido evolui autonomamente devido à sua natureza dinâmica. As políticas activas podem ser desenhadas para funcionar com requisitos dinâmicos, oferecendo regras para um espaço de intervenção potencialmente livre. Além disso, os sistemas de gestão de redes actuais normalmente têm uma base de dados centralizada onde são guardados todos os eventos que ocorreram, e sobre a qual podem actuar sistemas periciais. A arquitectura de políticas activas proposta é fortemente distribuída, pois colocam-se as políticas activas junto dos equipamentos geridos, por delegação. Esta aproximação garante tempos de resposta baixos, indispensáveis para assegurar os SLAs, reduzindo o tráfego de gestão na rede, aumentando a escalabilidade, flexibilidade e tolerância a falhas.

# CAPÍTULO 4

## ESTUDO DE UM CASO - REDE EMPRESARIAL

Neste capítulo ilustra-se a aplicação do modelo de políticas activas ao caso concreto de uma rede empresarial. Começa-se por introduzir o cenário, aplica-se o modelo de gestão ao cenário, explica-se como o cenário foi simulado, como funcionam as políticas activas para gestão da rede e os resultados obtidos. Por fim, avaliam-se os resultados obtidos.

### 4.1 CENÁRIO

As cada vez maiores exigências de qualidade de serviço por parte dos utilizadores levam a que os gestores de rede e sistemas tenham de se preocupar com requisitos de desempenho, fiabilidade e segurança. Em muitos casos pretende-se formalizar estas garantias em acordos de níveis de serviço (SLAs).

Neste capítulo retoma-se o exemplo de uma rede empresarial, já introduzido na secção 3.2, onde se pretende analisar a QoS obtida nos processos de negócio de uma empresa. Estuda-se apenas o requisito que exige que as vendas demorem menos de 10 segundos. Este é um exemplo de uma garantia que se pretende determinística, uma vez que nunca deve ser excedida. No entanto, não sendo possível assegurá-la sob todas as circunstâncias, só será possível ter garantias do tipo melhor esforço.

Para manter o contrato, as políticas activas devem otimizar os parâmetros da rede e dos sistemas, melhorando o seu funcionamento, e reduzindo os atrasos. Adicionalmente, as políticas activas podem evitar que se entre numa zona de degradação do serviço, impedindo, quando tal for indispensável, que novos utilizadores entrem no sistema.

## **4.2 APLICAÇÃO DO MODELO - ARQUITECTURA**

Nesta secção descreve-se a realização da primeira fase da metodologia para a construção de um sistema de gestão por políticas activas, correspondente ao estudo preliminar do problema, conforme proposto anteriormente na secção 3.5.

### **4.2.1 Nível Processo de Negócio**

No nível processo de negócio, os parâmetros de QoS permitem avaliar a globalidade da qualidade de serviço oferecida aos clientes da empresa, facultando medidas do tempo de execução de serviços, débito dos serviços, disponibilidades, e custos.

As políticas activas neste nível analisam a QoS que está a ser obtida e realizam modificações no funcionamento das políticas activas dos níveis inferiores de forma a assegurar os níveis de QoS pretendidos com um custo mínimo.

Neste nível apenas se vai analisar o requisito de as vendas demorarem menos de 10 segundos. Se este valor for excedido, deveriam ser acordadas penalidades, o que sai fora do âmbito desta tese. Assim, as políticas activas têm de actuar quando o atraso se aproximar deste limite de 10 segundos, de modo a evitar que ele seja ultrapassado. Nas situações de funcionamento normal, as políticas activas apenas calculam a QoS, sem agir.

### **4.2.2 Nível Serviço**

No nível serviço, as políticas activas podem evitar que entre no sistema um número excessivo de clientes, como solução de último recurso para assegurar o contrato. Desta forma, pretende-se evitar que os servidores entrem numa zona de funcionamento degradado, prejudicando o desempenho do sistema e o serviço oferecido a todos os clientes. Uma desvantagem deste procedimento é que os clientes podem, ao ser-lhes barrada a entrada, não obter sequer nenhum serviço.

### **4.2.3 Nível Tarefa**

No nível tarefa, os parâmetros de QoS permitem avaliar o desempenho de cada servidor individual.



As políticas activas neste nível permitem gerir o funcionamento dos serviços individuais. Há a possibilidade de desactivar serviços não indispensáveis em alturas críticas, como, por exemplo, suspender temporariamente os *backups* (cópias de segurança) em curso. É também possível realizar a distribuição dos pedidos pelos vários servidores disponíveis, de forma a manter a carga o mais aproximadamente possível igual em todos os servidores.

O requisito de nível processo de negócio de uma venda demorar menos de 10 segundos é refinado em requisitos do nível tarefa tal como ilustrado na figura 4.1, em conformidade com o já exposto na secção 3.2 e na figura 3.2. O processamento de uma venda exige uma consulta dos estoques, seguida da emissão da factura da venda. No nível tarefa este processo exige os cinco passos ilustrados na figura 4.1, tendo-se atribuído a cada um os tempos indicados, com uma pequena folga.

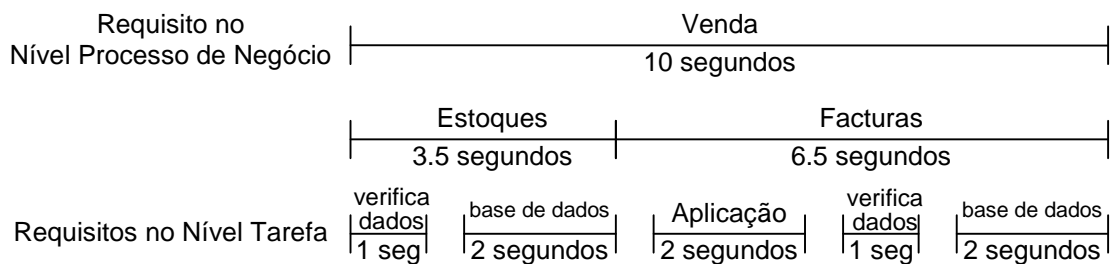


Figura 4.1: Refinamento do requisito do tempo de processamento de uma venda.

Para atingir os objectivos propostos, é necessário construir modelos aproximados para os parâmetros de QoS que representam o desempenho dos servidores, identificando os parâmetros dos objectos geridos que para eles contribuem.

Assim, tomando como exemplo o atraso de uma transacção na base de dados, a que corresponde um parâmetro de QoS do nível tarefa, constrói-se o modelo ilustrado na figura 4.2, onde o atraso de uma transacção na base de dados depende do atraso na rede e da carga na base de dados. Este modelo simplificado não quantifica a contribuição de cada parâmetro, apenas indica que o atraso médio de uma transacção na base de dados é tanto maior quanto maior for o atraso na rede ou a carga na base de dados.

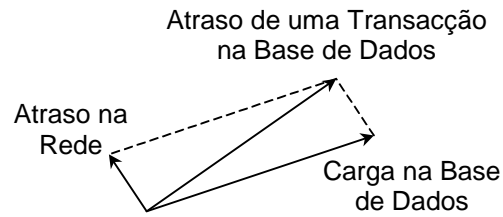


Figura 4.2: Contribuições para o atraso médio na base de dados.

Utilizando este modelo, é possível pensar o que poderá ser modificado quando o atraso de uma transacção na base de dados for excessivo, e a QoS se aproximar de níveis inaceitáveis. Como nenhum dos parâmetros indicados pode ser modificado directamente, torna-se necessário refinar o modelo, encontrando outros parâmetros que contribuam para a QoS e possam ser modificados. Na figura 4.3 ilustra-se um dos refinamentos possível, onde a carga da base de dados depende do número de pedidos dos utilizadores e dos *backups* da base de dados. Assim, conclui-se que uma forma de reduzir o atraso médio das transacções na base de dados é reduzir o número de pedidos dos utilizadores, o que é conseguido com as restrições impostas pelas políticas activas do nível serviço que fazem o controlo de acessos, ou com a política activa de nível tarefa que suspende os *backups* em alturas críticas.

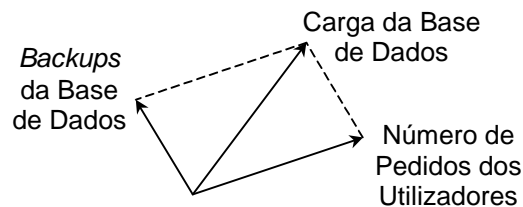


Figura 4.3: Contribuições para a carga na base de dados.

Note-se que, como alguns destes parâmetros pertencem a diferentes níveis de abstracção, torna-se necessário que as decisões de actuação sejam tomadas nos níveis superiores quando envolvam parâmetros desses níveis.

#### 4.2.4 Nível Rede e Sub-Rede

Nos níveis rede e sub-rede não se utilizou nenhuma política activa.

Seria possível medir os atrasos da rede como um parâmetro de QoS, e actuar sobre a configuração da rede. No entanto, numa rede local, a principal modificação que

poderia ser feita seria o aumento da sua capacidade que, como já foi focado anteriormente, não pode ser feita por políticas activas. Assim, neste caso, as políticas activas limitar-se-iam ao cálculo de parâmetros de QoS (tais como a carga, atrasos e erros na rede), e, possivelmente, à emissão de alarmes e sugestões para o planeamento da rede.

#### 4.2.5 Nível Equipamento

No nível equipamento, as políticas activas optimizam alguns parâmetros específicos de equipamentos particulares.

Como exemplo, toma-se a utilização de *caches* nos clientes da base de dados e da aplicação, que permitem optimizar as operações remotas, reduzindo a carga na rede e nos servidores, mas exigindo mais recursos de processamento e armazenamento nos clientes.

A figura 4.4 ilustra um modelo simplificado dos parâmetros que contribuem para o atraso médio obtido pelos clientes num acesso a um servidor. O atraso médio que os clientes sofrem será o atraso de um acesso ao servidor menos a melhoria obtida pela *cache*. Esta melhoria corresponde a uma consulta de uma *cache* local ao cliente, evitando-se o pedido ao servidor, caso a resposta já esteja armazenada na *cache*.

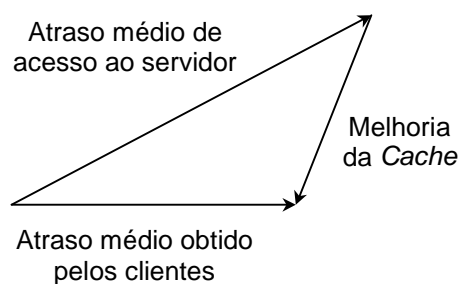


Figura 4.4: Contribuições para o atraso médio obtido pelos clientes.

Admitindo-se, como é habitual, que o tempo de acesso ao servidor é muito maior que o da consulta da *cache*, o benefício da utilização da *cache* é tanto maior quanto maior for a probabilidade de se encontrar a resposta já armazenada na *cache*. Geralmente, esta probabilidade aumenta com a dimensão da *cache*, pelo que é possível

utilizar uma política activa para gerir a dimensão da *cache* de forma a otimizar o desempenho dos serviços.

## 4.3 SIMULAÇÃO

Para testar as políticas activas, construiu-se uma simulação de uma rede empresarial. Nesta secção descreve-se o simulador, a rede simulada e os parâmetros de simulação utilizados.

### 4.3.1 O Simulador *Ptolemy*

Para simular uma rede empresarial utilizou-se o simulador *Ptolemy Classic* [Ptolemy], versão 0.7, desenvolvido pelo projecto Ptolemy do Departamento de Engenharia Electrotécnica e Computadores da Universidade da Califórnia em Berkeley, que está disponível gratuitamente.

O simulador *Ptolemy* é um simulador extensível, orientado para objectos, que suporta vários tipos de modelos de simulação. O simulador inclui um ambiente gráfico, chamado *piggi* (*Ptolemy interactive graphical interface*), que permite criar diagramas, configurá-los e simulá-los. Os principais modelos de simulação suportados são: eventos discretos, fluxos de dados síncronos, máquinas de estados finitas, e processos comunicantes. O simulador pode ser estendido criando novos objectos em C++ que podem ter interfaces em Tcl/Tk [Ousterhout 94].

Foi utilizado o modelo de simulação por eventos discretos. Neste modelo, toda a informação que passa entre objectos do simulador é um evento que corresponde a uma mudança no estado do sistema. Cada evento tem um instante de tempo associado, sendo o processamento dos eventos realizado por ordem cronológica. Cada objecto pode gerar destes eventos, com tempos dependentes dos tempos dos eventos processados nas suas entradas e da latência própria do objecto. Este é um modelo adequado para a simulação de redes de comunicação, sistemas de filas de espera, equipamentos, e redes de transporte. O simulador já inclui objectos neste modelo para gerar eventos, visualizar eventos, simular filas de espera, servidores e atrasos, medir intervalos e atrasos, testar condições lógicas, simular uma rede Ethernet, e realizar estatísticas simples, entre outros.

Para realizar a interface entre as políticas activas e o sistema simulado decidiu-se utilizar o protocolo SNMP. Para isso recorreu-se às extensões à linguagem Tcl para SNMP proporcionadas pelo Scotty [Schonwalder 95], que se integrou com a interface Tcl do *Ptolemy*.

Embora o *Ptolemy* seja um bom simulador, com uma extensa biblioteca de objectos que simulam vários tipos de equipamentos, especialmente no domínio do processamento de sinais, foi necessário criar um conjunto de novos objectos, bem como modificar alguns dos existentes, para melhor simular o sistema pretendido:

- Objecto MIB. Novo objecto criado para suportar a colocação de valores produzidos pelos objectos do simulador numa MIB para poderem ser lidos por SNMP, e para suportar a colocação como entrada para os objectos do simulador de valores de uma MIB escritos por SNMP.
- Objectos emissor e receptor Ethernet (`EthernetSend` e `EthernetRec`). Para implementar o algoritmo de retransmissão em caso de colisão que não era suportado pelo objecto emissor Ethernet já existente no simulador, e acrescentar uma fila no objecto receptor Ethernet para permitir receber novos pacotes antes de o anterior ter sido lido. Estes objectos simulam uma Ethernet sem fios, visto não haver ligação directa entre o emissor e os receptores, mas o funcionamento é idêntico a uma Ethernet por cabo.
- Objecto que gera eventos de acordo com um processo Poisson com taxa que pode ser modificada durante a simulação (`PoissonVar`). O objecto já existente no simulador apenas permite ter uma taxa fixa.
- Objecto que emula um servidor determinístico com partilha de processador (`PSServ`). O objecto já existente no simulador teve de ser corrigido, pois tinha um erro que causava problemas em algumas situações. O objecto funciona da seguinte forma: se chegarem eventos de entrada quando o servidor está livre, os eventos são atrasados o tempo de serviço nominal. Se chegarem quando o servidor está ocupado, o servidor é partilhado. Assim, os eventos mais antigos que ainda estejam em serviço são atrasados mais que o tempo de serviço nominal, bem como os novos.
- Objecto que emula a geração da resposta de um servidor (`Servidor`). Quando recebe um evento de entrada, gera um número configurável de eventos de saída com duração uniformemente distribuída dentro de um dado intervalo.

- Objecto que mede o tempo que decorreu entre os eventos nas suas duas entradas (TimeDiff).
- Objecto que conta eventos e regista o resultado em ficheiro (myCountEvents).
- Objecto que faz algumas estatísticas sobre os valores colocados na sua entrada, incluindo a média, desvio padrão, máximo e mínimo, e regista os resultados em ficheiro (myStatistics).

### 4.3.2 Topologia Cliente / Servidor

Para simular o processo de uma venda foi construída a topologia ilustrada na figura 4.5 tal como aparece no ambiente gráfico do simulador.

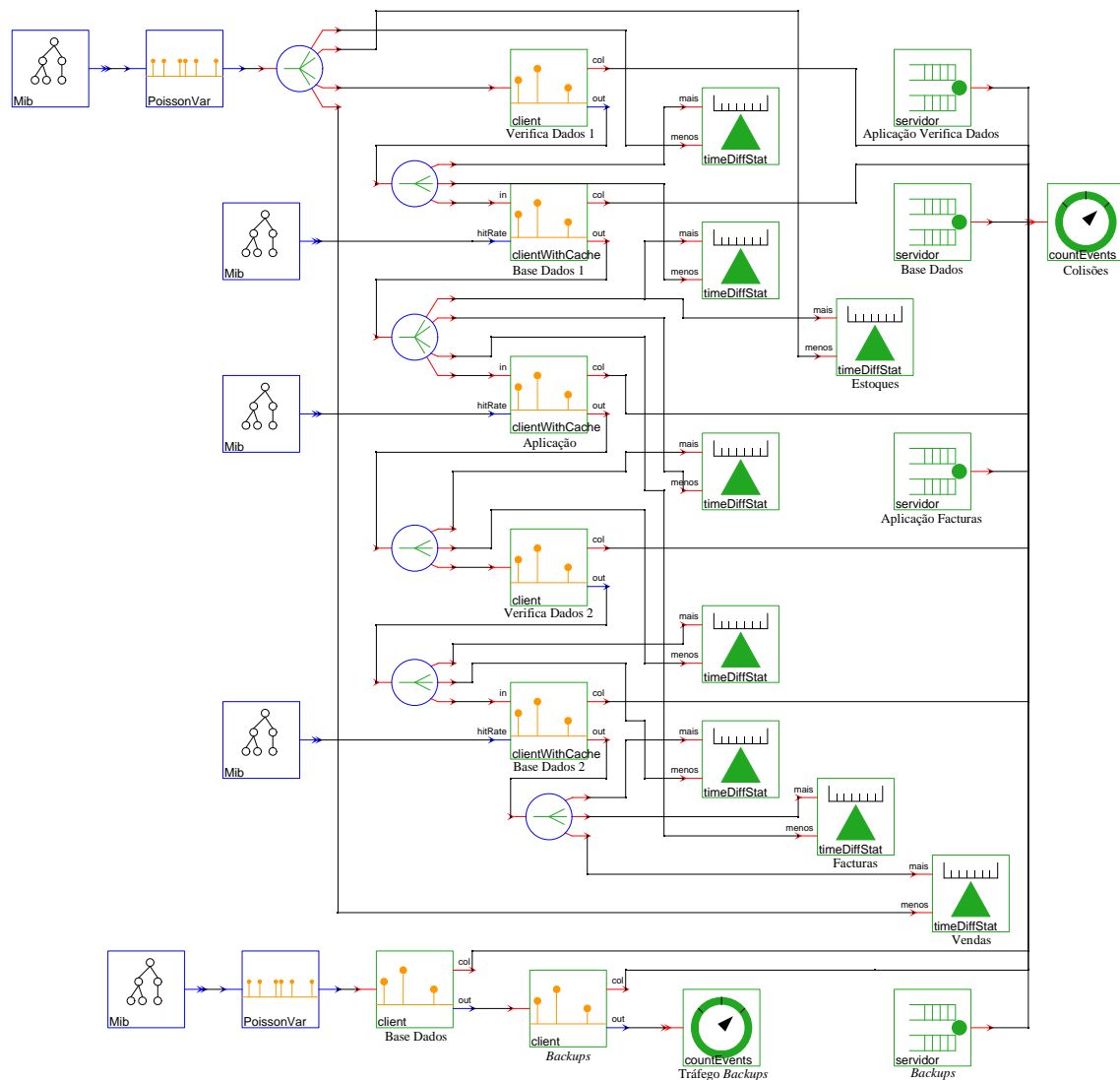


Figura 4.5: Topologia simulada representando os clientes e servidores.

O bloco `PoissonVar`, no canto superior esquerdo, faz a geração dos eventos correspondentes aos pedidos de operações de venda por parte dos clientes de acordo com uma taxa que pode ser configurada externamente através de uma MIB. Este pedido desencadeia no bloco `client` de cima a realização de um pedido ao servidor correspondente para verificar os dados, conforme o primeiro passo indicado anteriormente na figura 4.1. O bloco `client` corresponde a um diagrama cuja topologia está ilustrada na figura 4.6. Este actua enviando um pacote de pedido pela Ethernet para o servidor. Os blocos `Const` permitem configurar o pacote de dados, a duração da transmissão e o endereço do servidor. A resposta que chegar do servidor será colocada na saída do bloco `client`.

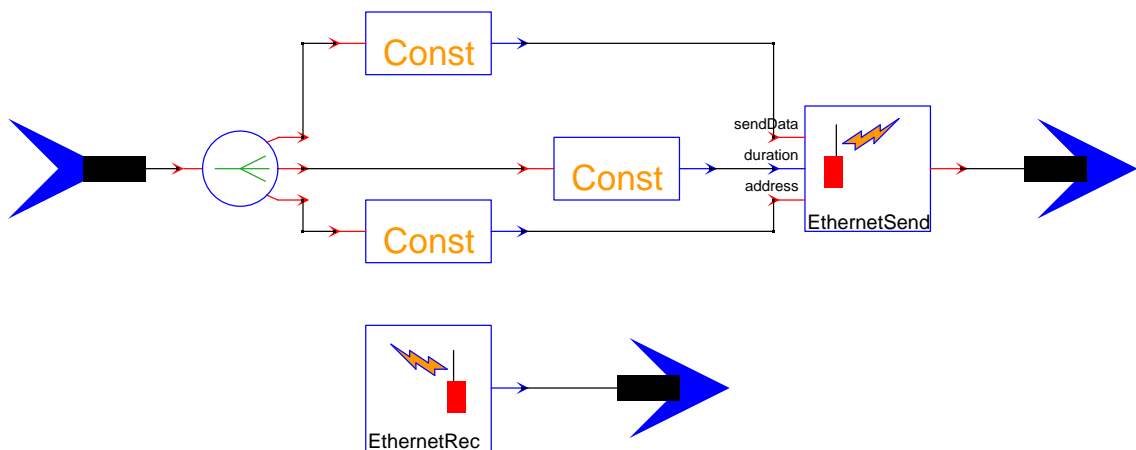


Figura 4.6: Topologia de um cliente.

O servidor corresponde ao diagrama cuja topologia está ilustrada na figura 4.7. Quando é recebido um pedido, este entra num objecto que emula um servidor determinístico com partilha de processador, seguindo-se um atraso constante, e sendo finalmente gerado um conjunto de pacotes de resposta para o cliente, cujo endereço provém do pacote recebido. No caso deste servidor apenas é gerado um único pacote de resposta.

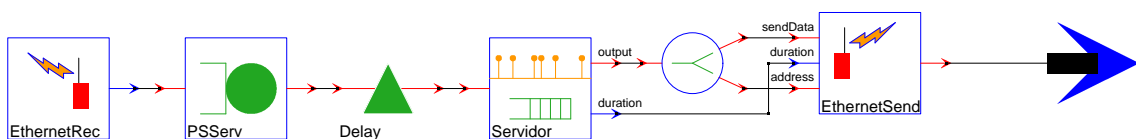


Figura 4.7: Topologia de um servidor.

Quando a resposta do servidor chegar ao cliente, o bloco `timeDiffStat` mede o tempo que esta demorou, desde o evento de início até ao evento da resposta, e o segundo bloco `client` é despoletado para realizar o passo seguinte da venda. O bloco `timeDiffStat` corresponde ao diagrama indicado na figura 4.8, sendo colocados numa MIB quatro valores correspondentes: ao número de pedidos satisfeitos; ao atraso do último pedido satisfeito; à média dos atrasos; e à sua variância.

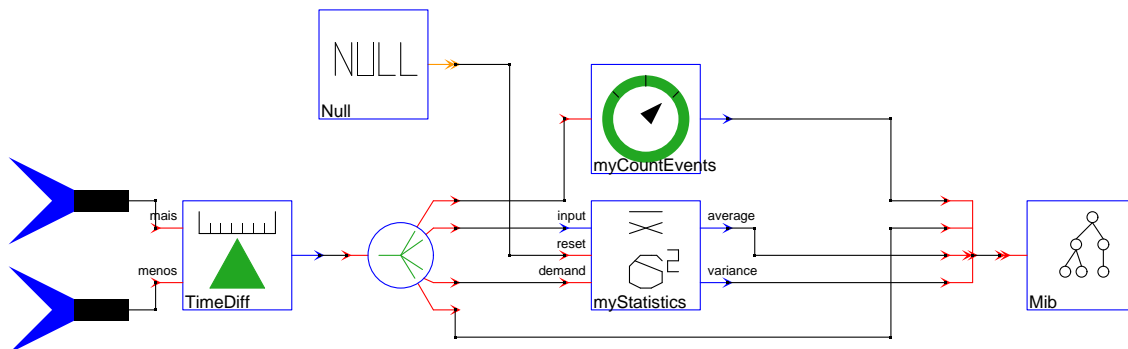


Figura 4.8: Topologia do bloco medidor de atrasos.

Os quatro clientes seguintes na figura 4.5 correspondem à simulação dos passos seguintes do processo de uma venda, de cima para baixo: consulta da base de dados de estoques, execução do processo de emissão da factura, verificação dos dados e consulta da base de dados para a emissão da factura. Os clientes que acedem às bases de dados e ao servidor de aplicações têm uma *cache* interna que pode evitar alguns pedidos ao servidor. A probabilidade de acerto na *cache* é configurável externamente por uma MIB.

Há blocos `timeDiffStat` adicionais para medir o tempo de execução do serviço de estoques (dois clientes de cima na figura 4.5), do serviço de emissão de facturas (três clientes seguintes), e do processo completo da venda (total dos cinco clientes).

Há três servidores do lado direito da figura 4.5 que correspondem, de cima para baixo, ao servidor da aplicação de verificação de dados, ao servidor da base de dados e ao servidor da aplicação de emissão de facturas.

Os dois clientes e servidor finais (na parte de baixo da figura 4.5) simulam o sistema de *backups* da base de dados, sendo que o primeiro faz um acesso à base de dados e o segundo ao servidor de *backups*. No caso do servidor de *backups* cada pedido



gera múltiplos pacotes de resposta. A taxa de pedidos de *backup* pode ser configurada através de uma MIB.

Todos os blocos que enviam pacotes para a Ethernet têm uma saída colisão, onde é colocado um evento em caso de colisão na Ethernet. O bloco `countEvents` no lado direito da figura 4.5 conta estes eventos em todos os emissores Ethernet, obtendo o número total de colisões na rede, que disponibiliza numa MIB.

### 4.3.3 Parâmetros de Simulação

Para simular a geração de pedidos de vendas no sistema, admitiu-se que chegavam novos pedidos de serviço a intervalos dados por uma distribuição exponencial com uma dada média. O objectivo de utilizar uma distribuição exponencial foi o de simular um processo sem memória.

Para simular a geração de pedidos de *backup* da base de dados, utilizaram-se os mesmos tipos de objectos no simulador, sendo, por isso, os intervalos entre pedidos dados por uma distribuição exponencial. À partida, pareceria mais natural ter os intervalos entre pedidos como constantes, ou uniformemente distribuídos dentro de um determinado intervalo. No entanto, na realidade os sistemas não são assim tão regulares, pelo que pareceu mais simples e geral utilizar a distribuição exponencial também neste caso.

O dimensionamento dos servidores foi feito conforme indicado na tabela 4.1, tendo-se admitido que a Ethernet era de 10 Mbps.

|                                    | <b>Servidor de Base de Dados</b> | <b>Servidor de Aplicação</b> | <b>Servidor de Verificação de dados</b> | <b>Servidor de Backups</b> |
|------------------------------------|----------------------------------|------------------------------|---|----------------------------|
| <b>Tamanho dos Pedidos</b>         | 125 bytes                        | 125 bytes                    | 125 bytes                               | 125 bytes                  |
| <b>Tamanho Médio da Resposta</b>   | 5 Kb                             | 25 Kb                        | 5 Kb                                    | 128 Kb                     |
| <b>Capacidade de Processamento</b> | 50 Pedidos/s                     | 25 Pedidos/s                 | 100 Pedidos/s                           | 100 Pedidos/s              |
| <b>Atraso Constante</b>            | 0.5 s                            | 0.5 s                        | 0.5 s                                   | 0.0 s                      |

Tabela 4.1: Dimensionamento dos servidores.

Admitiu-se que inicialmente as *caches* estão todas desligadas.

Admitiu-se que o cliente do serviço de *backups*, quando activo, faz em média 3 pedidos por segundo, o que corresponde a utilizar cerca de um terço da largura de banda disponível na rede local.

Relativamente ao dimensionamento da taxa de pedidos dos clientes do serviço de vendas, importa estudar o que acontece com diferentes taxas. Pode-se olhar, a título de exemplo, para a figura 4.9 que ilustra o tráfego médio horário num período de uma semana para um segmento de rede local Ethernet a 10 Mbps. A média foi feita recolhendo dados de uma sonda RMON2 [RFC 2819][RFC 2021] pelo período de 16 semanas, num segmento de rede correspondente a parte do 5º andar do Inesc ID Lisboa. Nesta figura, o dia 1 corresponde ao domingo e o dia 7 ao sábado.

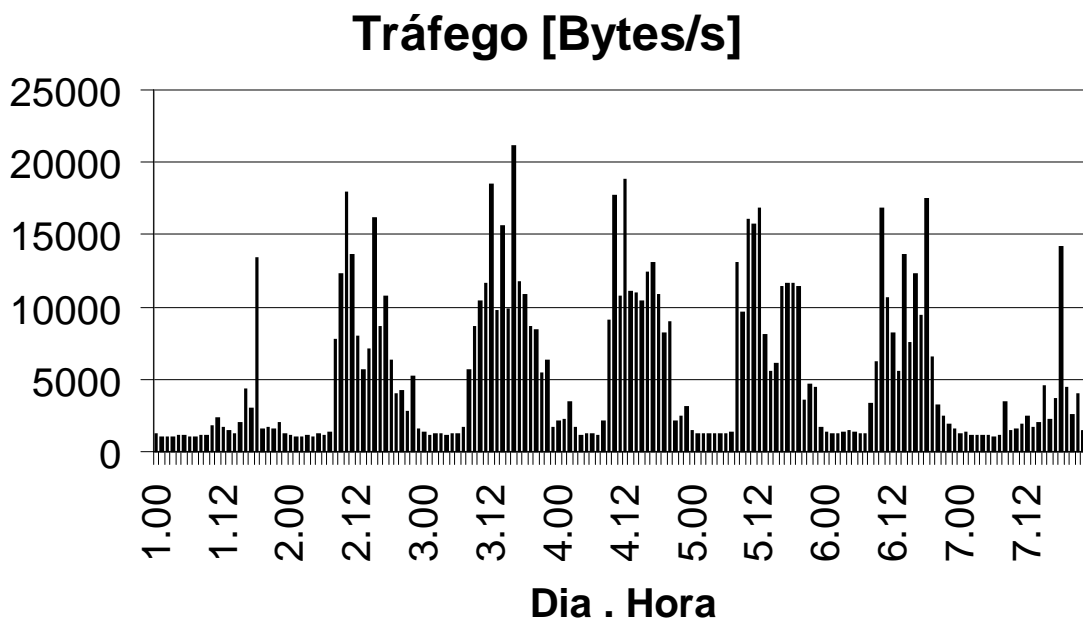


Figura 4.9: Tráfego médio horário num segmento de rede local no período de uma semana.

Outro exemplo está ilustrado na figura 4.10 que mostra o número médio de acessos em cada hora ao servidor [www.ist.utl.pt](http://www.ist.utl.pt). A média foi feita recolhendo dados ao longo do mês de Julho de 2000.

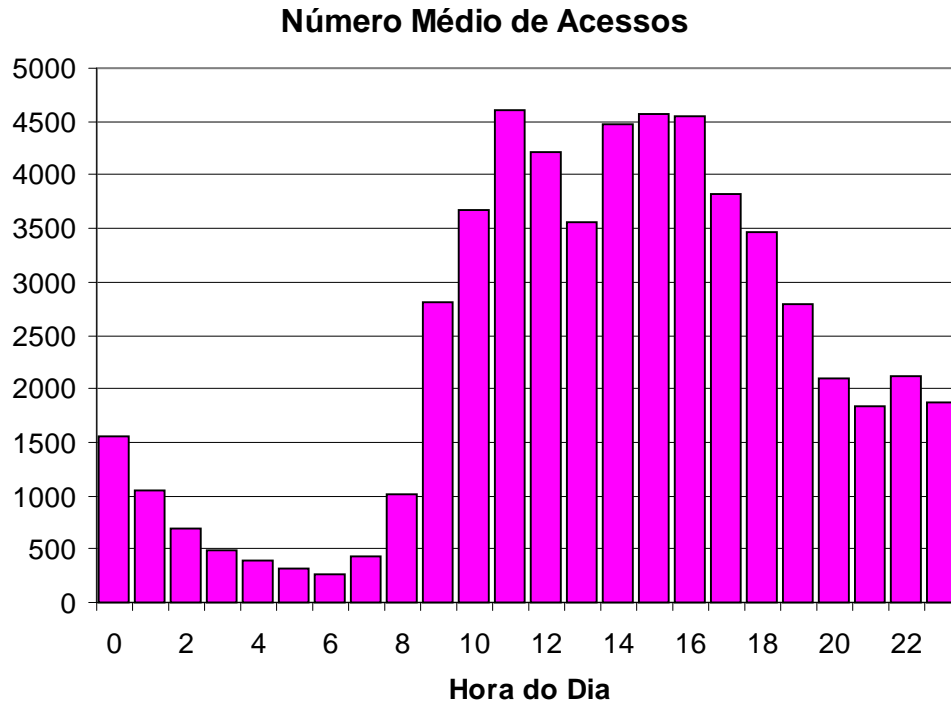


Figura 4.10: Número médio de acessos em cada hora ao servidor [www.ist.utl.pt](http://www.ist.utl.pt).

Da comparação da figura 4.9 e figura 4.10 conclui-se que o tráfego varia muito ao longo do dia, e um pouco consoante o dia. Há horas, normalmente de noite, em que o tráfego é muito baixo, e horas em que o tráfego é mais intenso. Assim, importa analisar a evolução do sistema com o aumento do tráfego, visto que o sistema deverá ser dimensionado pelo tráfego máximo que se pretende servir com uma dada qualidade de serviço.

#### 4.3.4 Discussão

O simulador realizado tem algumas simplificações relativamente à realidade.

O simulador só simula a transmissão de pacotes Ethernet, não simulando toda a pilha de protocolos por cima da Ethernet, nem acessos a outros serviços indispensáveis, tais como servidores de nomes.

A Ethernet simulada tem as seguintes aproximações:

- tempo de propagação nulo.
- quando há múltiplas transmissões em simultâneo, a primeira a ser simulada ocupa o meio de transmissão, e as outras detectam-no como ocupado.

- quando se tenta transmitir e o meio de transmissão está ocupado, actua-se como se tivesse havido colisão, sem perturbar a transmissão em curso.
- não há tamanho máximo de pacote, nem informação de controle nos pacotes, nem erros de transmissão.

As transferências de informação por SNMP das MIBs são feitas através de uma interface de rede da máquina onde corre o simulador (normalmente a interface local), e não através da rede simulada, pelo que não consomem nenhuma largura de banda à rede simulada. Além disso, a comunicação entre políticas activas é feita por chamadas a procedimentos dentro do simulador, não havendo *polling* que causaria um ligeiro atraso e algum consumo de largura de banda da rede simulada.

Pensa-se que estas simplificações não invalidam os resultados obtidos, apenas fazem com que a rede simulada seja mais eficiente do que uma rede real. O único caso que pode causar problemas ao passar para o mundo real é o caso da emissão de notificações de problemas que, se demorarem muito, podem causar uma actuação que já não corresponde à situação existente na rede. Assim, é necessário ter algum cuidado com os tempos de resposta, de forma a evitar oscilações no funcionamento dos sistemas. No caso da simulação realizada, a detecção de problemas é instantânea (visto não haver *polling*), o que corresponderia, num sistema real, a ter as políticas activas no mesmo local onde são medidos os tempos de atraso de cada pedido, tal como proposto anteriormente na secção 3.7. No entanto, as actuações sobre o sistema não são instantâneas. Por exemplo, caso se decida suspender os *backups*, o pedido em curso será totalmente processado, pelo que a suspensão só é efectiva depois disso, o que já corresponde a um funcionamento real.

Relativamente ao protocolo cliente/servidor, também se admitiu sempre um protocolo simplificado ao máximo, com um único pacote de pedido e um único pacote de resposta, à excepção do servidor de *backups* que responde com múltiplos pacotes pequenos. Tal não corresponde directamente a nenhum sistema real, mas é uma boa aproximação do comportamento típico das interacções cliente/servidor. Para ter uma simulação mais exacta da realidade seria necessário implementar uma simulação de protocolos reais nos clientes e servidores. Em todo o caso, a ênfase deste trabalho está em concluir sobre a exequibilidade das políticas activas e da arquitectura de gestão, não sendo importantes as pequenas diferenças de interacção dependentes de um protocolo

concreto. Não se pretendeu, assim, no cenário deste capítulo simular nenhum protocolo particular.

Relativamente ao funcionamento das *caches* nos clientes, admitiu-se que se podia configurar a taxa de acerto nas *caches* e não as suas dimensões, o que corresponde a uma aproximação que não é possível de obter directamente na realidade. Para não tornar a simulação irrealista, admitiu-se que a taxa de acertos na *cache* dos clientes da base de dados podia ser apenas: 0% (*cache* desligada), ou 10% (*cache* ligada); enquanto que na *cache* dos clientes da aplicação podia ser apenas: 0% (*cache* desligada), 10% (*cache* na dimensão normal), ou 20% (*cache* na dimensão máxima).

Relativamente ao modelo de funcionamento interno do servidor, admitiu-se, conforme ilustrado anteriormente na figura 4.7, que o servidor introduzia um atraso constante nos dados, mais um atraso variável consoante a carga do servidor, de acordo com um modelo de servidor determinístico com partilha de processador. Pensa-se que é um modelo simples, mas verídico, para o que realmente acontece num servidor com recursos limitados, que são divididos pelos vários pedidos que estão a ser servidos em simultâneo.

Pensa-se que estas simplificações ao funcionamento dos clientes e servidores não invalidam os resultados obtidos, visto que se procurou simular uma certa capacidade de processamento de pedidos e uma certa carga na rede, tornando a situação simulada realista.

Seria possível construir um modelo teórico do sistema simulado, mas tal não é feito, pois o modelo não corresponderia a um sistema real genérico, mas ao caso particular do exemplo construído. Novamente, o objectivo deste trabalho foi avaliar a metodologia para resolver um problema de gestão com políticas activas, e não estudar detalhes de um sistema particular tomado como exemplo.

## **4.4 APLICAÇÃO DO MODELO - POLÍTICAS ACTIVAS**

A figura 4.11 ilustra as políticas activas identificadas para cada nível, e como elas se relacionam entre si. Nesta secção descreve-se detalhadamente o funcionamento das várias políticas activas, correspondendo à segunda fase da metodologia para a construção de um sistema de gestão por políticas activas, conforme proposto

anteriormente na secção 3.5. Esta arquitectura e as principais políticas activas foram descritas em [Pereira 99b].

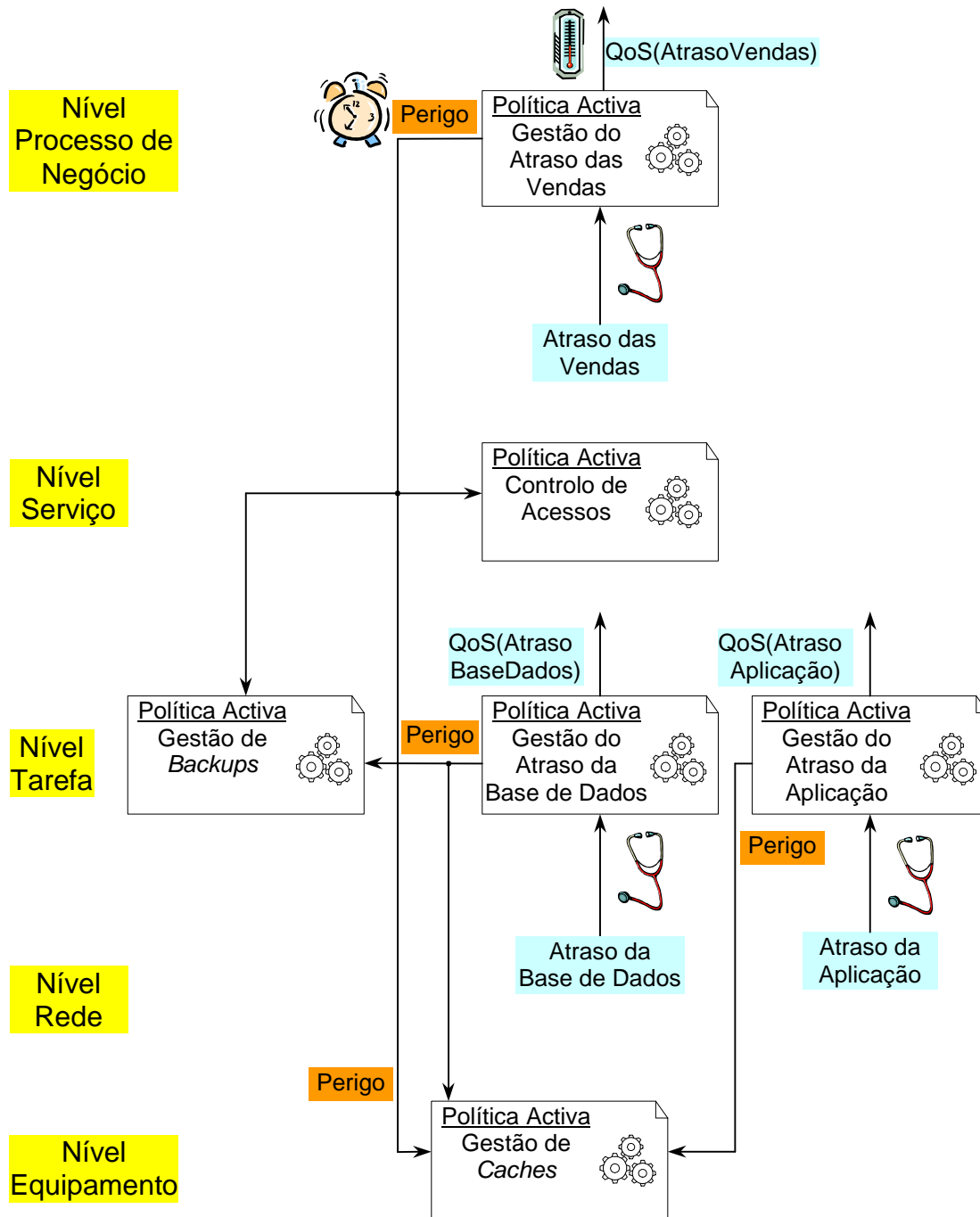


Figura 4.11: Políticas activas utilizadas e as suas relações.

#### 4.4.1 Política Activa de Gestão do Atraso das Vendas

A política activa de gestão do atraso das vendas actua localmente à interface com o cliente, do lado do fornecedor de serviço, monitorizando o tempo que as vendas demoram. Esta política disponibiliza em parâmetros de QoS o tempo de atraso da última venda, e o tempo de atraso médio das vendas, calculado através de uma média móvel de peso exponencial (EWMA) que reduz o peso de cada medida a metade por cada período de 3 segundos.

Para fazer cumprir o requisito de as vendas demorarem menos de 10 segundos, esta política activa envia notificações de perigo de violação do contrato quando o atraso se aproxima do limite imposto. Para isso, é utilizada a regra `threshold`, emitindo-se uma notificação de “perigo” para as políticas activas de gestão de *backups* e gestão de *caches* quando se atinge 80% do limite imposto para o atraso de uma venda, e uma notificação de “perigo” para a política activa de gestão de controlo de acessos quando se atinge 90% do limite imposto para o atraso de uma venda. Por outro lado, se o atraso das vendas baixar abaixo de 50% do limite imposto, é enviada uma notificação de “fora de perigo” para todas as políticas activas para onde se enviou previamente uma notificação de “perigo”.

Esta política activa funciona como último recurso para evitar violações do contrato, esperando-se que as políticas activas de níveis mais baixos resolvam problemas pontuais com maior antecedência.

#### 4.4.2 Política Activa de Gestão de Controlo de Acessos

A política activa de gestão de controlo de acessos actua localmente à interface com o cliente, do lado do fornecedor de serviço, podendo inibir temporariamente o acesso ao sistema de novos utilizadores para evitar que se entre numa zona de degradação de funcionamento.

Esta política activa não calcula nenhum parâmetro de QoS, só actuando quando recebe notificações da política activa de nível superior que gere o atraso das vendas.

O número máximo de novos clientes por unidade de tempo que podem ser admitidos no sistema é ajustado com a regra `decrease`, a intervalos de 10 segundos. Quando é recebida uma notificação de “perigo”, o número máximo de clientes é decrementado de 2 unidades a cada 10 segundos. Quando é recebida uma notificação de

“fora de perigo”, o número máximo de clientes é incrementado de 2 unidades a cada 10 segundos, até atingir o valor normal.

Na simulação realizada, em vez de se impor um máximo ao número de clientes, barrando a entrada a novos clientes que excedam o máximo, baixa-se a taxa de geração de novos clientes.

#### **4.4.3 Política Activa de Gestão do Atraso da Aplicação**

A política activa de gestão do atraso da aplicação actua localmente à interface com o cliente da aplicação, tendo-se admitido que estava do lado do fornecedor de serviço, em conjunto com a *cache*. Esta política activa monitoriza o tempo que os pedidos à aplicação demoram, disponibilizando em parâmetros de QoS o tempo de atraso do último pedido e o tempo de atraso médio dos pedidos, calculado através de uma média móvel de peso exponencial (EWMA), que reduz o peso de cada medida a metade por cada período de 3 segundos.

Para fazer cumprir o requisito de os pedidos à aplicação demorem menos de 2 segundos, esta política activa envia notificações de perigo de violação do contrato quando o atraso médio se aproxima do limite imposto. Para isso, é utilizada a regra *threshold*, emitindo-se uma notificação de “perigo” para a política activa de gestão de *caches* quando se atinge 50% do limite imposto para o atraso médio dos pedidos. Quando o atraso médio dos pedidos baixar abaixo de 40% do limite imposto, é enviada uma notificação de “fora de perigo”. Além disso, é também enviada uma notificação de “perigo grave” para a política activa de gestão de *caches* quando se atinge 70% do limite imposto para o atraso médio dos pedidos. Finalmente, se o atraso médio dos pedidos baixar abaixo de 55% do limite imposto, é enviada uma notificação de “fora de perigo grave”.

#### **4.4.4 Política Activa de Gestão do Atraso da Base de Dados**

A política activa de gestão do atraso da base de dados actua localmente ao cliente da base de dados, monitorizando o tempo que os pedidos à base de dados demoram. Esta política disponibiliza em parâmetros de QoS o tempo de atraso do último pedido, e o tempo de atraso médio dos pedidos, calculado através de uma média



móvel de peso exponencial (EWMA) que reduz o peso de cada medida a metade por cada período de 3 segundos.

Para fazer cumprir o requisito de os pedidos à base de dados demorarem menos de 2 segundos, esta política activa envia notificações de perigo de violação do contrato quando o atraso médio se aproxima do limite imposto. Para isso, é utilizada a regra *threshold*, emitindo-se uma notificação de “perigo” para a política activa de gestão de *backups* quando se atinge 50% do limite imposto para o atraso médio dos pedidos. Quando o atraso médio dos pedidos baixar abaixo de 40% do limite imposto, é enviada uma notificação de “fora de perigo”. Além disso, é também enviada uma notificação de “perigo” para a política activa de gestão de *caches* quando se atinge 70% do limite imposto para o atraso médio dos pedidos. Finalmente, se o atraso médio dos pedidos baixar abaixo de 55% do limite imposto, é enviada uma notificação de “fora de perigo” para a política activa de gestão de *caches*.

#### **4.4.5 Política Activa de Gestão de Backups**

A política activa de gestão de *backups* actua localmente à aplicação que realiza os *backups*, podendo inibir temporariamente a realização de *backups* da base de dados para reduzir a carga na base de dados e na rede.

Esta política activa não calcula nenhum parâmetro de QoS, só actuando quando recebe notificações de outras políticas activas.

Sempre que recebe uma notificação de “perigo”, suspende os *backups*, terminando apenas o bloco em curso. Só recomeça os *backups* quando tiver recebido notificações de “fora de perigo” de todas as políticas activas de onde recebeu anteriormente uma notificação de “perigo”.

#### **4.4.6 Política Activa de Gestão de Caches**

A política activa de gestão de *caches* existe em duas versões, uma que actua localmente ao cliente da base de dados e outra que actua localmente ao cliente da aplicação. Em ambos os casos, gere a dimensão da *cache* do cliente.

Esta política activa não calcula nenhum parâmetro de QoS, só actuando quando recebe notificações de outras políticas activas.

Sempre que a política activa de gestão da *cache* do cliente da base de dados recebe uma notificação de “perigo”, activa a utilização da *cache*. Quando tiver recebido notificações de “fora de perigo” de todas as políticas activas de onde recebeu anteriormente uma notificação de “perigo”, desactiva a utilização da *cache*.

Se a política activa de gestão da *cache* do cliente da aplicação receber uma notificação de “perigo”, activa a utilização da *cache*. Se receber uma notificação de “perigo grave”, configura a *cache* para a dimensão máxima. Quando tiver recebido notificações de “fora de perigo grave” de todas as políticas activas de onde recebeu anteriormente uma notificação de “perigo grave”, reconfigura a *cache* novamente para a dimensão normal. E quando tiver recebido notificações de “fora de perigo” de todas as políticas activas de onde recebeu anteriormente uma notificação de “perigo”, desactiva a utilização da *cache*.

## 4.5 RESULTADOS OBTIDOS

Realizaram-se simulações do sistema com diferentes cargas de pedidos, com e sem a utilização das políticas activas. As figuras 4.12 e 4.13 ilustram, respectivamente, o atraso médio e máximo de uma venda em função da carga de pedidos. Em cada um dos gráficos há uma curva, marcada com “♦”, correspondente à situação com as políticas activas a funcionar, e uma curva, marcada com “×”, correspondente à situação sem as políticas activas a funcionar.

Como é natural, à medida que a carga de pedidos aumenta, o sistema fica mais carregado, demorando mais tempo a servir os pedidos. Verifica-se que, quando não são utilizadas políticas activas, o sistema entra em degradação a partir de uma carga de cerca de 17.5 pedidos/s. Com a utilização das políticas activas, consegue-se manter o atraso máximo abaixo do limite imposto de 10 segundos para cargas até 20 pedidos/s, deixando de se obter o efeito de degradação.

Na figura 4.14 ilustram-se os atrasos (em segundos) obtidos nas vendas, nos acessos à base de dados e à aplicação, para um período de 90 segundos, com uma carga de 20 pedidos/s. Na política activa de gestão do atraso das vendas é utilizado directamente o atraso de cada venda, tal como aparece no gráfico correspondente da figura 4.14, de forma a poder-se reagir rapidamente a situações que possam causar uma

violação do contrato. Por outro lado, nas restantes políticas activas é utilizada a média móvel, tal como aparece nos outros dois gráficos da figura 4.14, de forma a filtrar pequenas perturbações. Os instantes em que as políticas activas actuaram, correspondendo aos limiares indicados anteriormente, estão também assinalados na figura.

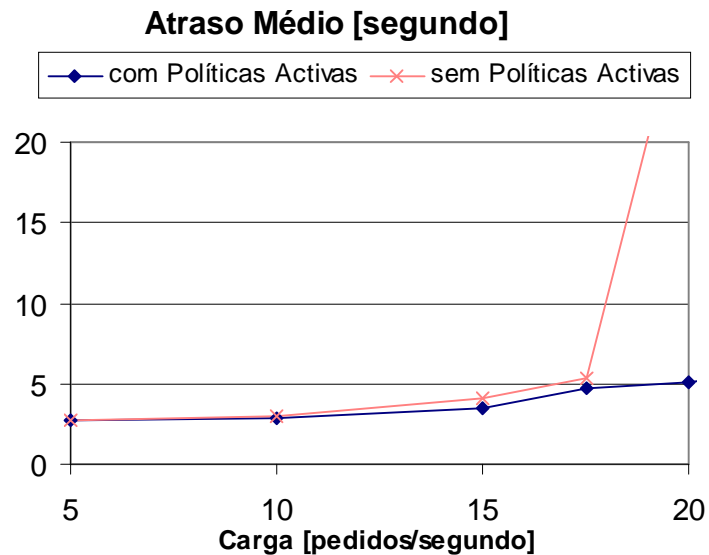


Figura 4.12: Comparação do atraso médio de uma venda em função da carga de pedidos nas situações com e sem as políticas activas.

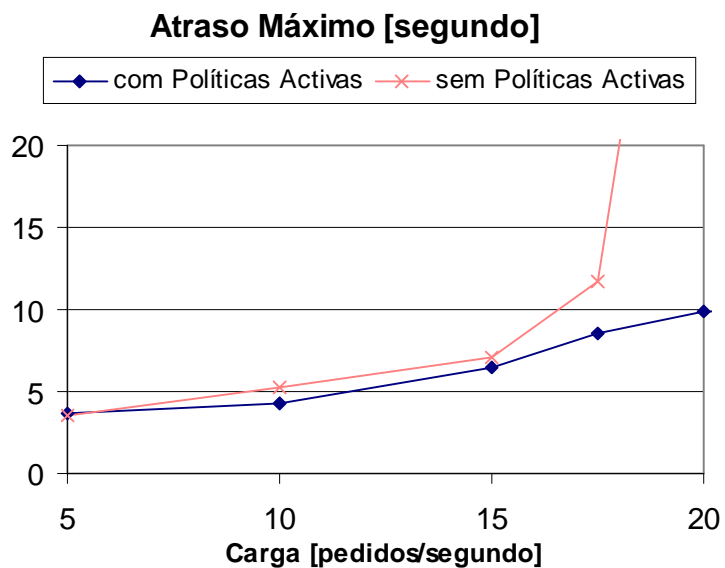


Figura 4.13: Comparação do atraso máximo de uma venda em função da carga de pedidos nas situações com e sem as políticas activas.

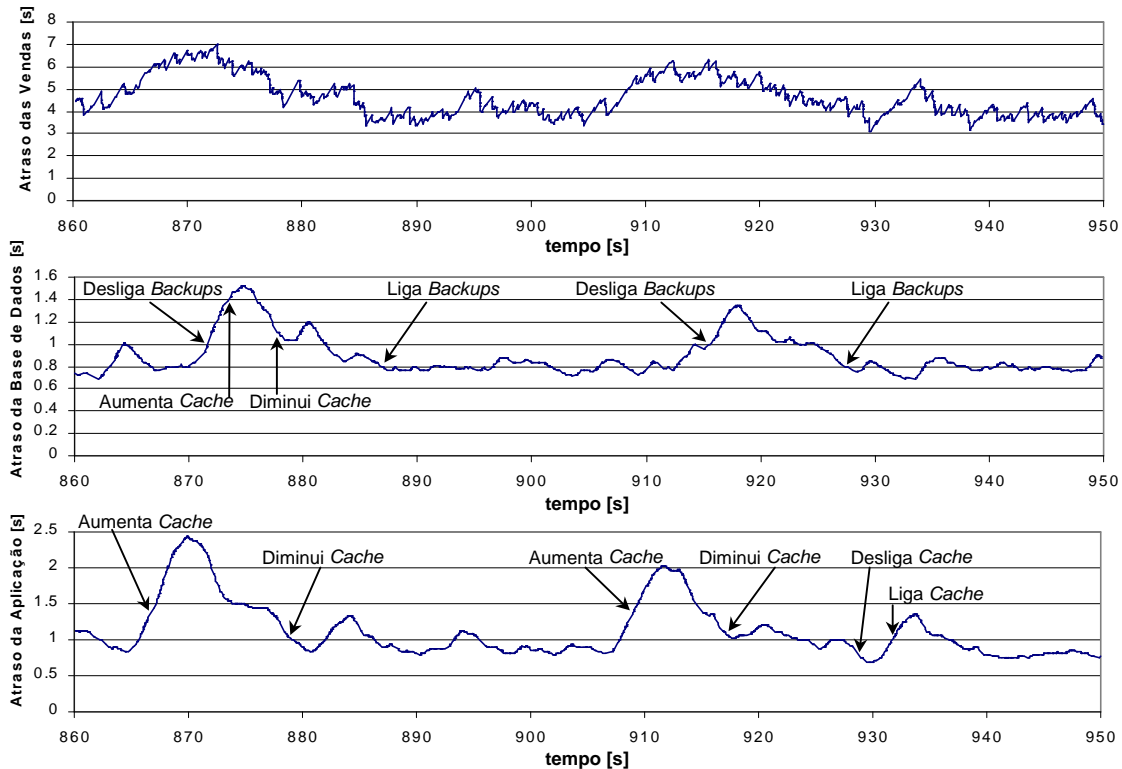


Figura 4.14: Resultados simulados do atraso com as políticas activas.

Nota-se que, embora o atraso da aplicação exceda o limite imposto de 2 segundos por algum tempo, o contrato de alto nível é respeitado. Além disso, com a carga de pedidos imposta, para esta simulação, o atraso das vendas nunca atinge 9 segundos, correspondente a 90% do limite imposto no contrato, pelo que nunca é barrado o acesso a novos clientes. Assim, verifica-se que, neste caso, basta actuar nos dois níveis mais baixos de políticas activas para assegurar o contrato de alto nível. No entanto, para cargas de pedidos superiores, torna-se necessário actuar a um nível mais alto, barrando progressivamente o acesso a alguns clientes, de forma a evitar a degradação do sistema e assegurar o contrato de alto nível. Por esta razão, os gráficos das figuras 4.12 e 4.13 param numa carga de 20 pedidos/s.

## 4.6 AVALIAÇÃO

Na secção 3.5.4 apresentaram-se alguns critérios com o objectivo de avaliar a eficiência, justiça, distribuição e convergência de um sistema de gestão. Nesta secção avalia-se o cenário estudado de acordo com cada um deles em maior pormenor.

A eficiência traduz a aproximação ao valor ideal. Verificou-se que foi possível fazer cumprir o requisito especificado no contrato de alto nível, embora em alguns casos os contratos de níveis inferiores não sejam respeitados. O preço a pagar, para que isto seja possível, é o acesso a novos utilizadores poder ser bloqueado, de forma a evitar que se entre numa situação de degradação do sistema. No entanto, como o tráfego numa rede é caracterizado por frequentes rajadas, por vezes correlacionadas, será natural esperar que nem sempre se consiga atingir os objectivos pretendidos actuando apenas na gestão. Este problema é estudado em [Bernardo 02], onde é proposta uma solução com agentes móveis para responder a picos correlacionados de pedidos.

A convergência mede a rapidez e a suavidade com que se chega ao resultado pretendido. Do ponto de vista de rapidez, procurou-se ter uma resposta rápida das várias políticas activas, a situações de falha, ou de falha iminente, e uma resposta lenta, uma política activa de cada vez, para minimizar as perturbações, no sentido de regresso à normalidade. Ou, em termos de QoS, procurou-se melhorar rapidamente a QoS sempre que é detectada a sua degradação, e reduzi-la lentamente quando é melhor do que o estritamente necessário, de acordo com o algoritmo genérico de ajuste de parâmetros do sistema descrito anteriormente na figura 3.10. Pelo facto de as políticas activas actuarem perto dos objectos geridos, minimiza-se o seu tempo de reacção, e reduz-se ao mínimo a possibilidade de estar a actuar em sentido contrário ao devido, maximizando-se assim as garantias de convergência.

No caso apresentado neste capítulo, os limiares de actuação utilizados foram escolhidos manualmente. Desta escolha depende a eficiência e a convergência do sistema. Como esta escolha pode nem sempre ser fácil, na secção 5.5.2 é analisado um método para ajustar dinamicamente os limiares de actuação.

A justiça mede o grau com que os recursos são equitativamente repartidos pelos utilizadores da mesma classe. Como os utilizadores individuais não são distinguidos pelo sistema, obtêm, em média, uma QoS muito semelhante, dependendo da carga na altura em que fazem pedidos.

Do ponto de vista da distribuição, pretende-se minimizar a informação de gestão transmitida na rede. Esta foi uma preocupação constante ao longo da concepção do sistema de gestão. De facto, ao colocar políticas activas, que podem actuar com uma certa independência, junto dos objectos geridos, permite-se eliminar parte do tráfego de gestão. Na realidade, numa implementação real deste cenário, as políticas activas

estariam todas na máquina que tem a aplicação de interface com o cliente, à excepção da política activa de gestão de *backups*, que estaria na máquina onde os *backups* são realizados. As situações em que as políticas necessitam de trocar informação com outras políticas são apenas: situações de excepção, caso em que é necessário enviar alarmes; e situações em que as políticas de nível superior necessitam de ajustar algum parâmetro nas políticas de nível inferior, caso que é relativamente raro. Assim, para este cenário, só seria necessário transmitir informação de gestão na rede para suspender e reactivar a aplicação que realiza os *backups*, sendo por isso o tráfego correspondente muito pouco significativo.

## 4.7 CONCLUSÕES

O exemplo simples descrito neste capítulo mostra como se pode avaliar a qualidade de serviço que os utilizadores estão a obter. Além disso, mostra-se como, controlando um reduzido número de parâmetros chave no sistema gerido, as políticas activas podem melhorar o desempenho do sistema e evitar que se entre numa zona de degradação de funcionamento.

As ideias aplicadas no cenário deste capítulo podem ser aplicadas a muitos outros cenários. Conjugando actuações em diferentes níveis de abstracção, tais como gestão de controlo de acessos, gestão de tráfego na rede, balanceamento de carga e optimização de parâmetros, pode-se atingir os objectivos pretendidos de gestão de alto nível.

# **CAPÍTULO 5**

## **ESTUDO DE UM CASO - FORNECEDOR DE SERVIÇO INTERNET**

Neste capítulo ilustra-se a aplicação do modelo de políticas activas ao caso concreto de um fornecedor de serviço Internet que utiliza uma rede com tecnologia de serviços diferenciados. Começa-se por fazer uma introdução aos serviços diferenciados, explicando apenas os conceitos necessários para compreender o cenário estudado, pois pretende-se focar a perspectiva de gestão. Depois aplica-se o modelo de gestão ao cenário, explica-se a forma como o cenário foi simulado, o funcionamento das políticas activas para gestão da rede e os resultados obtidos. Finalmente, estuda-se como é possível ajustar automaticamente os limiares das políticas activas de forma a fazer cumprir com maior exactidão os SLAs, e avaliam-se os resultados obtidos.

### **5.1 CENÁRIO**

#### **5.1.1 Introdução ao Problema**

Com o crescimento da Internet, e o aumento das capacidades gráficas e de cálculo dos computadores pessoais, têm aparecido novos serviços baseados em aplicações sobre rede, tais como aplicações multimédia, voz sobre IP, distribuição de vídeo, comércio electrónico, trabalho cooperativo, ensino assistido por computador, entre outros [Leiner 00]. Muitas destas novas aplicações exigem requisitos de qualidade de serviço que a Internet actual não suporta. Exemplos destes requisitos são: baixo

atraso extremo a extremo, baixo *jitter*<sup>1</sup>, baixa taxa de perda de pacotes e débito garantido.

Como a Internet actual oferece o mesmo tipo de serviço, melhor esforço, para todos os utilizadores, tratando todos os tipos de tráfego da mesma forma, tornou-se necessário introduzir novos mecanismos que permitissem classificar o tráfego e tratá-lo de forma diferente dentro da rede, consoante o tipo de serviço pretendido pelo utilizador. Na realidade, o serviço de tipo melhor esforço caracteriza-se por tratar todos os pacotes de igual forma, descartando pacotes como forma de aliviar a congestão, o que é insuficiente para suportar os novos serviços que têm requisitos de tempo real. Um serviço do tipo melhor esforço comporta-se bem apenas no caso de a rede estar levemente carregada, tratando mal todos os utilizadores no caso de congestão. Para obter um serviço de boa qualidade quando se utiliza uma filosofia de melhor esforço, a única solução é aumentar a largura de banda o suficiente para evitar as situações de congestão, resultando normalmente numa subutilização de recursos.

Uma primeira forma de suportar diferentes qualidades de serviço foi a introdução de Serviços Integrados (**IntServ** [RFC 1633]), definindo um modelo de serviços Internet que inclui o serviço melhor esforço, serviço de tempo real e partilha controlada das linhas. Para isso, os *routers* (encaminhadores) têm de suportar diferentes qualidades de serviço, no que se designa por Controlo de Tráfego. O controlo de tráfego é implementado com três componentes: o controlo de admissão, o classificador de pacotes e o mecanismo de serviço de pacotes. Além disso, é necessário utilizar um protocolo de estabelecimento de reservas de recursos.

O controlo de admissão implementa um algoritmo de decisão que determina se podem ser oferecidas garantias de qualidade a um novo fluxo de dados, sem prejudicar os fluxos já existentes. O classificador mapeia os pacotes numa das classes de tráfego. Todos os pacotes de uma mesma classe têm o mesmo tratamento pelo mecanismo de serviço. O mecanismo de serviço gere o envio de pacotes através de um conjunto de filas de espera e dos procedimentos associados.

O protocolo de reserva de recursos utilizado é o **RSVP** (*Resource ReSerVation Protocol* [RFC 2205]). A figura 5.1 ilustra, simplificada, como o protocolo funciona para estabelecer uma reserva unidireccional de recursos entre um emissor e um

---

<sup>1</sup> Optou-se por utilizar o termo inglês. O *jitter* corresponde à variação do atraso.



receptor. O emissor envia uma mensagem PATH caracterizando o tráfego em termos de limites de largura de banda, atraso e *jitter*. Todos os *routers* ao longo do percurso seguido pelos pacotes guardam informação sobre o fluxo assim definido. O receptor envia uma mensagem RESV que, além da caracterização do tráfego atrás indicada, inclui o protocolo de transporte, porto, identificador do fluxo de dados e o tipo de serviço IntServ requerido, que pode ser:

**Garantido** [RFC 2212]: pretende emular um circuito virtual dedicado, oferecendo limites rígidos (demonstrados matematicamente) sobre os atrasos e largura de banda disponibilizados.

**Carga Controlada** [RFC 2211]: é equivalente a melhor esforço sob condições de pouca carga. Embora não dê garantias absolutas, utiliza controlo de admissão de capacidade para que o serviço requerido seja oferecido mesmo em situações de sobrecarga da rede.

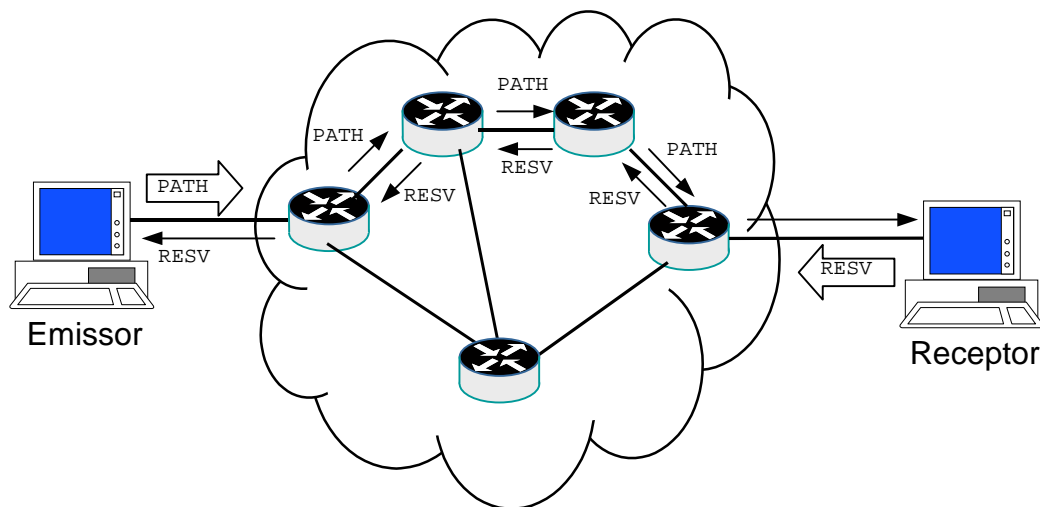


Figura 5.1: Exemplo de operação do protocolo RSVP para estabelecer uma reserva de recursos entre um emissor e um receptor.

Desta breve explicação ressalta um problema deste modelo. Para o RSVP funcionar, necessita de reservar recursos por cada fluxo de dados, em cada *router*, ao longo de todo o percurso. Numa rede em que podem coexistir simultaneamente milhões de ligações, isto não parece ser uma boa solução. Além disso, o RSVP exige que as reservas sejam periodicamente refrescadas, no que se designa por funcionamento em *soft-state*. O *soft-state* tem algumas vantagens: não exige comunicação fiável entre

entidades pares; é robusto, tolerando a perda de pacotes de refrescamento; permite possíveis mudanças do encaminhamento dos pacotes na rede; e permite possíveis mudanças da qualidade de serviço que pode ser oferecida. No entanto, o *soft-state* prejudica ainda mais a escalabilidade do RSVP.

Um outro modelo que pretende solucionar estes problemas é o modelo de Serviços Diferenciados, introduzido na subsecção seguinte.

### 5.1.2 Introdução aos Serviços Diferenciados

O modelo de Serviços Diferenciados (*Differentiated Services*, ou **DiffServ** [RFC 2475]) pretende obter diferenciação de serviços na Internet, de forma escalável, agregando tráfego em classes de tráfego. Os pacotes são classificados e marcados apenas ao entrar na rede, sendo encaminhados até ao seu destino consoante o comportamento especificado para a classe de tráfego respectiva. Desta forma, os *routers* do núcleo da rede apenas vêem o tráfego agregado em cada classe, deixando toda a complexidade de condicionamento do tráfego (classificação, marcação, policiamento e suavização dos pacotes) para os *routers* da periferia da rede.

Actualmente estão definidos dois novos tipos de serviços para além do existente:

**Envio Expedito** (*Expedited Forwarding*, ou **EF** [RFC 2598]): pretende fornecer um serviço do tipo circuito virtual, com uma certa largura de banda garantida, baixa taxa de descarte de pacotes, baixo atraso e baixo *jitter*. O utilizador contrata uma largura de banda (de pico), que não pode ser excedida, sendo todo o tráfego em excesso descartado. O fornecedor de serviço deve assegurar que este é o tráfego de maior prioridade, exceptuando o tráfego de controlo da rede, e não vender mais serviço do que a rede suporta. A vantagem para o fornecedor de serviço, relativamente a um circuito virtual, é que, no caso de o utilizador não utilizar toda a largura de banda que contratou, esta fica disponível para ser utilizada por outros tipos de serviço.

**Envio Assegurado** (*Assured Forwarding*, ou **AF** [RFC 2597]): pretende fornecer um serviço em que os pacotes que não excedam o ritmo negociado são entregues com elevada probabilidade, estando os pacotes que excedem esse ritmo sujeitos a uma maior probabilidade de descarte. O utilizador contrata uma largura de banda correspondente ao ritmo médio que utiliza, podendo opcionalmente especificar um determinado ritmo de pico. Este tipo de serviço pode incluir várias classes, estando

actualmente definidas quatro classes, cada uma com os seus próprios recursos atribuídos na rede. Cada classe de tráfego tem três níveis de probabilidade de descarte, sendo normal [RFC 2698] utilizar o nível com menor probabilidade de descarte para o tráfego abaixo do ritmo médio negociado; o nível intermédio para o tráfego que exceda o ritmo médio, mas não exceda o ritmo de pico; e o nível de maior probabilidade de descarte para o tráfego que exceda o ritmo de pico.

**Melhor Esforço** (*Best Effort*, ou **BE**): corresponde ao serviço já existente, sem quaisquer garantias, que é seleccionado por omissão.

A distinção de tratamento dos pacotes é conseguida pela utilização do campo DS no cabeçalho dos pacotes IP, com o código de serviço diferenciado **DSCP** (*Differentiated Services Codepoint* [RFC 2474]), que vem substituir os campos tipo de serviço (TOS) de IPv4 e classe de tráfego de IPv6. A estrutura deste campo está ilustrada na figura 5.2.

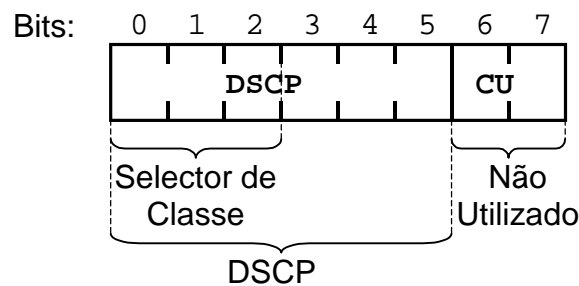


Figura 5.2: Estrutura do campo DS dos pacotes IP.

Para preencher correctamente este campo torna-se necessário realizar funções de condicionamento de tráfego ilustradas na figura 5.3. A gestão deste condicionamento de tráfego é feita através de um acordo de condicionamento de tráfego (*Traffic-Conditioning Agreement*, ou TCA), que especifica todas as regras utilizadas. O TCA inclui regras para: classificar pacotes em perfis de tráfego; definir as características dos perfis, como sejam ritmos médios, ritmos de pico e tamanhos de rajadas; e ainda definir o que fazer ao tráfego fora de perfil.

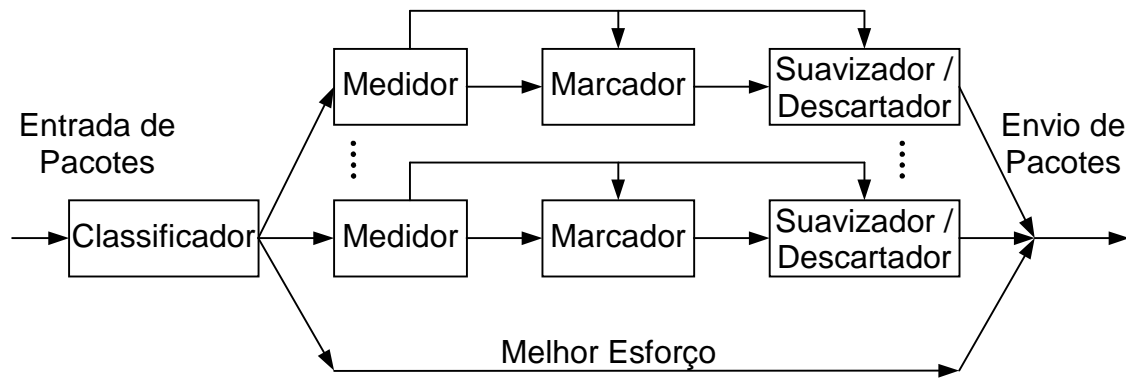


Figura 5.3: Funções de condicionamento de tráfego.

O **Classificador** de pacotes (*classifier*) é um mecanismo utilizado para seleccionar a classe a atribuir a um fluxo de tráfego. Podem-se pensar em vários métodos de classificação:

- O utilizador selecciona a classe pretendida de entre as disponíveis.
- A aplicação do utilizador selecciona a classe mais adequada para cada fluxo de pacotes, ou pacote.
- A rede selecciona a classe apropriada. Esta selecção pode ser feita com base na origem ou destino dos pacotes, na aplicação (porto TCP/UDP), na hora actual e no contrato com o utilizador.
- Combinações dos anteriores.

O **Medidor** de tráfego (*meter*) mede o ritmo de transmissão de dados, com o objectivo de influenciar a operação do Marcador e do Suavizador/Descartador. O medidor também pode ser utilizado para fazer taxação.

O **Marcador** de pacotes (*marker*) marca os pacotes com o campo DSCP adequado, podendo depender da indicação de um medidor. Quando modifica a marcação existente, diz-se que remarcou os pacotes.

O **Suavizador** (*shaper*) pode atrasar pacotes com o objectivo de reduzir os picos, tornando o fluxo de pacotes conforme com um dado perfil. O **Descartador** (*dropper*) é o caso particular de um suavizador com *buffer* (memória tampão) nulo, que descarta pacotes com o objectivo de tornar o fluxo de pacotes conforme com um dado perfil.

A figura 5.4 ilustra os grupos de classes de tráfego, as classes de tráfego, os campos DSCP correspondentes às diferentes probabilidades de descarte de cada classe

(alta, média, baixa) e um possível mapeamento para mecanismos de envio de tráfego (*Per-Hop Behaviour*, ou PHB). É habitual designar as marcações correspondentes aos níveis de probabilidade de descarte por cores, utilizando-se vermelho para a alta probabilidade de descarte, o amarelo para a média probabilidade de descarte, e o verde para a baixa probabilidade de descarte. Cada PHB corresponde a um comportamento de envio de pacotes na rede. Um PHB é o meio pelo qual um nó atribui recursos à transmissão de um fluxo de tráfego agregado. Normalmente, o PHB corresponde à atribuição de uma percentagem da largura de banda disponível e de *buffers* para as filas de espera de pacotes. No caso de os recursos atribuídos a um PHB não estarem a ser totalmente utilizados, as sobras ficam disponíveis para serem utilizadas pelos outros PHBs. Convém notar que o mapeamento de DSCP para PHB pode ser de um para um, ou de N para um.

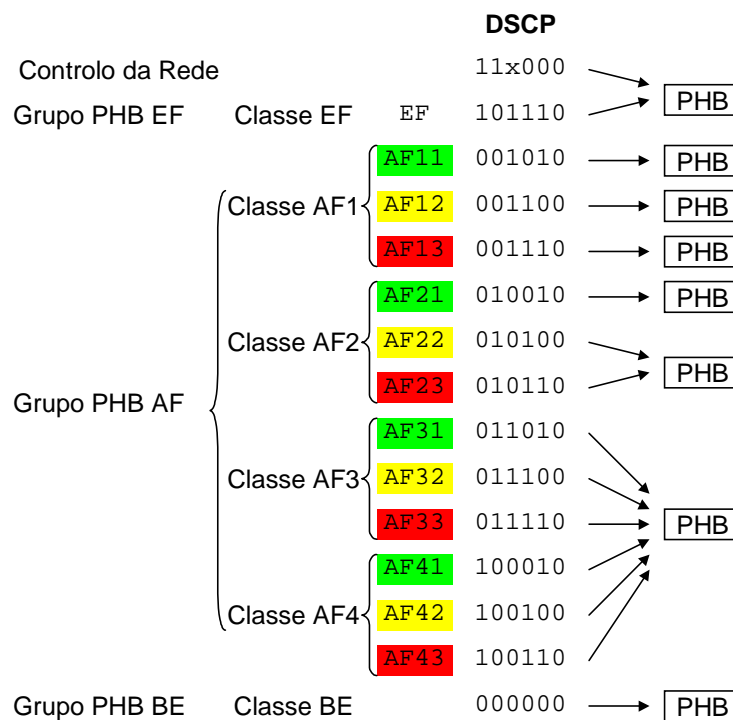


Figura 5.4: Classes DiffServ, Codepoints e PHBs.

Até agora focou-se o problema de como conseguir implementar serviços diferenciados na rede, pela marcação de pacotes em várias classes de tráfego, e dedicação de recursos da rede a cada classe de tráfego. Referiu-se que tem de haver um método de classificação dos pacotes numa das classes de tráfego, mas, como facilmente se pode depreender, havendo um limite para os recursos atribuídos a cada classe de

tráfego, também tem de haver um limite para o tráfego que lá é colocado, de forma a poder garantir a qualidade de serviço. Se tal não for feito, obtém-se um serviço semelhante ao melhor esforço.

Em [RFC 2638] é proposto um mecanismo de **Corretores de Largura de Banda** (*Bandwidth Broker*, ou BB) que são agentes locais a uma organização, ou fornecedor de serviço Internet (*Internet Service Provider*, ou ISP), que mantêm informação sobre a largura de banda atribuída em cada instante, interpretam políticas de atribuição de largura de banda, e respondem a pedidos de obtenção de largura de banda. Pensa-se que numa primeira fase de implementação de serviços diferenciados as reservas de recursos sejam estáticas, de longa duração, estabelecidas manualmente. Numa segunda fase, as reservas seriam dinâmicas, de duração flexível, estabelecidas automaticamente recorrendo ao corretor de largura de banda.

A figura 5.5 ilustra o funcionamento do estabelecimento de uma reserva entre um emissor e um receptor utilizando corretores de largura de banda. Neste exemplo, as reservas são estáticas, sendo feitas antecipadamente, pelos valores indicados nos *routers* de entrada de cada rede na figura, utilizando-se troca de mensagens entre corretores de largura de banda para maior flexibilidade. Por exemplo, a organização ORG1 pode enviar 100 Kbps de tráfego marcado para o fornecedor ISP1. As tabelas da figura 5.5 indicam as políticas dos ISPs, e apresentam os valores após a reserva estar concluída.

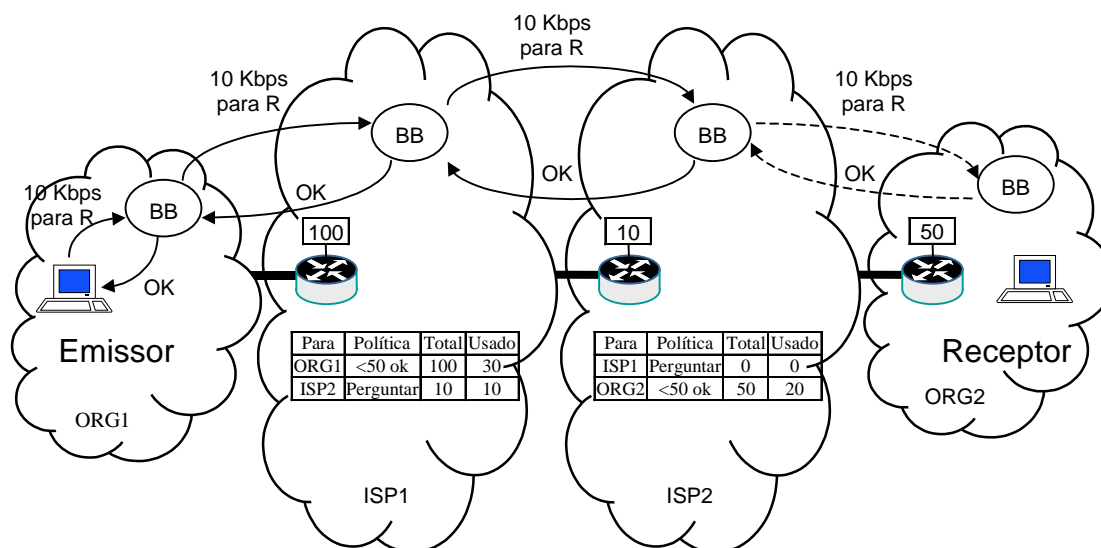


Figura 5.5: Estabelecimento de uma reserva através do Corretor de Largura de Banda.

O Emissor de ORG1 pretende reservar 10 Kbps para o Receptor em ORG2. Para isso, ele pede ao seu BB, que autentica o pedido, e verifica se há 10 Kbps disponíveis na reserva feita nessa direcção. Como há, ele passa o pedido ao BB de ISP1, dizendo que pretende usar 10 Kbps para este fluxo de dados. O BB de ISP1 autentica a mensagem, e verifica a base de dados, concluindo que tem 10 Kbps de reserva não utilizada para ISP2. A política é que o ISP1 deve sempre informar (perguntar) ao ISP2 quando quer utilizar parte da reserva. O ISP2 autentica a mensagem, verifica a sua base de dados, e descobre que há 40 Kbps da reserva para ORG2 não utilizados, sendo a política não informar ORG2 quando se pretende usar parte da reserva (“<50 ok”, sendo 50 a reserva total). As setas a traço interrompido representam a transacção implícita correspondente, ficando 30 Kbps não utilizados, e 20 Kbps utilizados, tal como indicado na figura 5.5. Agora cada BB pode enviar uma mensagem “ok” como resposta a este pedido, permitindo que o Emissor envie para o Receptor, mas não vice-versa.

Com reservas estáticas entre BBs, se em algum dos passos não houver reserva disponível, é devolvida ao Emissor uma mensagem indicando que a reserva não foi estabelecida. Caso se utilizem reservas dinâmicas, os BBs têm de actualizar as reservas existentes nos *routers* de entrada à medida que aceitam novas reservas. Uma das formas possíveis para transmitir a informação de reservas é utilizar o protocolo RSVP.

### 5.1.3 Discussão

Neste cenário optou-se por utilizar serviços diferenciados como plataforma para suportar qualidade de serviço na rede. Há outras tecnologias alternativas e ainda outras tecnologias que podem ser conjugadas com serviços diferenciados para este fim, não sendo abordadas neste trabalho. Um exemplo de uma tecnologia destas é a norma IEEE 802.1p, que permite a definição de prioridades no tráfego Ethernet.

Muitos dos mecanismos necessários para o correcto funcionamento de uma rede com serviços diferenciados ainda estão em desenvolvimento, estando alguns problemas ainda à espera de resposta. Por exemplo, é pouco claro como fazer controlo de admissão de forma eficiente numa rede sem ligação lógica, e como conceber protocolos para reservas entre domínios. Estes problemas têm sido sentidos no projecto Internet2 [Teitelbaum 99], que pretende desenvolver uma rede com qualidade de serviço para suporte de aplicações avançadas baseadas em rede.

Da apresentação feita, nota-se que o controlo de admissão pode variar desde uma opção muito simples em que todos os pedidos são admitidos, obtendo-se um serviço de melhor esforço, passando por uma opção em que só é consultado o BB local, sendo as reservas entre domínios estáticas, até uma solução mais complexa em que todas as reservas são dinâmicas extremo a extremo. Esta última opção aproxima-se da solução prevista por RSVP, exceptuando que apenas os BBs e os *routers* da periferia da rede mantêm informação sobre fluxos individuais, enquanto os *routers* do núcleo da rede mantêm apenas informação sobre o tráfego agregado.

Na realidade, para obter garantias rígidas de qualidade é necessário fazer reservas extremo a extremo, pois nada impede que haja vários nós em fornecedores independentes a trocar tráfego com o mesmo nó destinatário, causando congestão local junto deste. Mas, por outro lado, não é eficiente fazer reservas extremo a extremo para milhões de ligações, muitas delas de curta duração, como acontece na Internet actual.

Convém ainda ter presente que um modelo para o serviço de rede tem de ser independente das aplicações. De facto, há 20 anos, a aplicação dominante na rede era execução remota de programas, há 10 anos era correio electrónico e transferência de ficheiros, hoje é a *Web*, e dentro de 10 anos pode ser outra totalmente diferente, por exemplo videofone, ou realidade virtual.

É ainda de referir que a introdução de conceitos de qualidade de serviço não cria largura de banda. Assim, se alguns utilizadores obtêm melhor serviço, outros obterão pior serviço. O objectivo é evitar ter de aumentar a largura de banda para todos, de maneira a que as aplicações que não podem tolerar congestão possam funcionar correctamente. Assim, a mesma largura de banda pode ser repartida de forma mais eficiente, de forma a que os diferentes tipos de aplicações vejam uma rede com a qualidade adequada ao seu funcionamento.

## 5.2 APLICAÇÃO DO MODELO - ARQUITECTURA

Os serviços fornecidos são regulados por acordos [Pereira 00b]. Um acordo de utilizador (*Customer Level Agreement*, ou CLA) especifica o serviço de rede a fornecer ao cliente pelo ISP. Um acordo de níveis de serviço (*Service Level Agreement*, ou SLA) especifica de um ponto de vista técnico os requisitos pretendidos por cada aplicação.



Finalmente um acordo de condicionamento de tráfego (*Traffic-Conditioning Agreement*, ou TCA) especifica o tratamento dos pacotes em cada equipamento.

A figura 5.6 ilustra as várias entidades envolvidas no fornecimento de um serviço Internet. As linhas a tracejado representam interfaces de fornecimento de serviço. O utilizador subscreve um serviço Internet ao fornecedor de serviço. Por outro lado, o utilizador recorre a aplicações para transmitir dados pela rede. A rede por sua vez está suportada em diversos equipamentos. O fornecedor de serviço recorre a mecanismos de gestão de rede para configurar a rede, de forma a suportar os serviços com a QoS contratada com o utilizador.

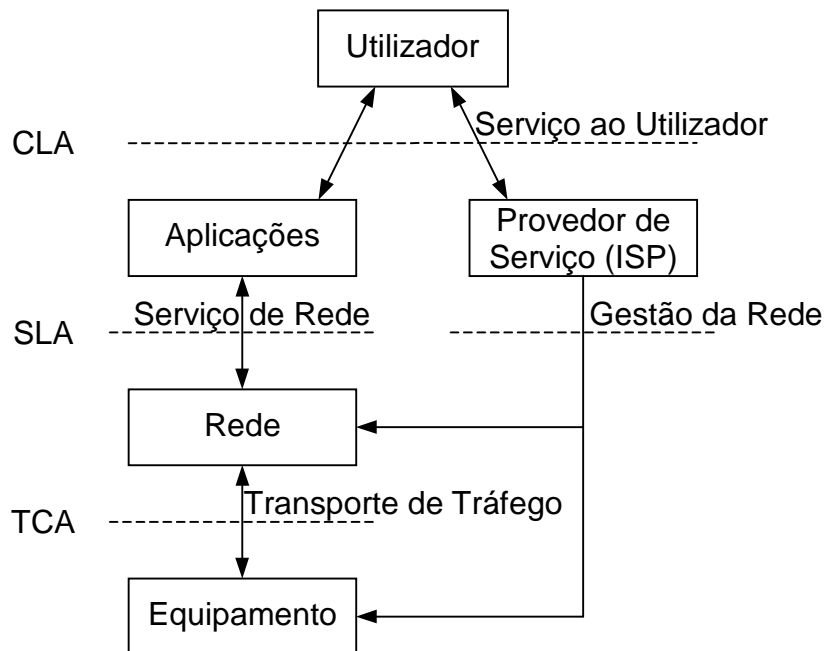


Figura 5.6: Serviços e acordos.

Nesta secção descrevem-se os vários níveis do modelo de gestão, começando pelos requisitos e o refinamento correspondente, seguindo-se os parâmetros de qualidade de serviço utilizados, a forma como as políticas activas actuam e as variáveis que controlam. Este estudo preliminar do problema corresponde à realização da primeira fase da metodologia para a construção de um sistema de gestão por políticas activas, conforme proposto anteriormente na secção 3.5.

### 5.2.1 Nível Processo de Negócio

No nível processo de negócio os requisitos são especificados de uma forma muito geral, sendo depois refinados nos outros níveis de abstracção para requisitos directamente implementáveis.

São definidas várias classes de serviço a oferecer aos clientes.

Classe **Ouro**: esta classe pretende emular um circuito virtual. É a classe com o serviço de melhor qualidade (menor atraso e *jitter*), sendo negociado um débito de pico que é assegurado. Esta classe é a mais adequada para aplicações de transmissão de voz ou vídeo.

Classe **Prata**: esta é uma classe intermédia entre a Ouro e a Bronze em termos de qualidade. É negociado um ritmo base que é assegurado, podendo o ritmo transmitido ser superior, mas sem ser assegurado. Esta classe é a mais adequada para aplicações interactivas, ou em que se pretenda ter uma rede rápida.

Classe **Bronze**: o serviço nesta classe é melhor esforço, não havendo nenhuma garantia para o utilizador. Esta classe é a mais adequada para aplicações de baixa prioridade, ou quando se pretender minimizar os custos.

Os clientes podem contratar uma, ou mais classes, marcando o seu tráfego, ou deixar que o operador seleccione a classe mais adequada a cada tipo de tráfego, negociando um perfil de mapeamento do seu tráfego nas classes de serviço.

Neste nível, a gestão da rede faz a monitorização do tráfego global da rede, dos valores cobrados aos clientes, do custo das linhas, dos SLAs cumpridos ou violados e do lucro obtido. Desta monitorização são produzidas indicações para o planeamento da rede. Em alguns casos poder-se-á fazer o ajuste automático de parâmetros de políticas de níveis inferiores.

### 5.2.2 Nível Serviço

Os requisitos das várias classes de serviço são refinados neste nível.

Classe **Ouro**: esta classe tem o seu tráfego classificado como EF. A disponibilidade do serviço deve ser elevada, devendo haver uma muito baixa taxa de ligações recusadas. Deve ter um atraso médio inferior ao especificado num SLA. O débito de pico negociado deve ser sempre garantido, sendo o tráfego em excesso descartado. Esta classe não deve ter descarte de pacotes dentro do débito negociado, a

menos que haja falha de linhas. Todos os parâmetros devem ser melhores do que na classe Prata.

Classe **Prata**: esta classe tem o seu tráfego classificado como AF. A disponibilidade do serviço deve ser grande, devendo haver uma baixa taxa de ligações recusadas. Deve ter um atraso médio inferior ao especificado num SLA. O débito negociado deve ser sempre garantido, sendo o tráfego em excesso transmitido, se a carga da rede o permitir. Esta classe deve ter uma taxa de descarte de pacotes muito baixa para ritmos inferiores ao negociado, a menos que haja falha de linhas. Todos os parâmetros devem ser melhores do que na classe Bronze.

Classe **Bronze**: esta classe tem o seu tráfego classificado como BE, não havendo nenhuma garantia para o utilizador.

No nível serviço é gerido o mapeamento do tráfego nas classes de tráfego. A marcação do tráfego pode ser feita do lado do cliente, ou do lado do fornecedor de serviço. O cliente pode marcar o tráfego de acordo com a aplicação particular que está a utilizar, com a prioridade que cada utilizador pretende ter, ou com os custos que está disposto a suportar. A marcação do tráfego também pode ser efectuada pelo operador da rede de acordo com o contrato realizado com o cliente, ou para utilizadores que não marquem o seu tráfego. O mapeamento do tráfego nas classes de tráfego é feito por regras que especificam qual a classe de tráfego a utilizar consoante a aplicação utilizada (porto e protocolo de transporte), o utilizador (endereço ou rede de origem ou de destino), hora actual, ou a marcação original.

Caso a marcação do tráfego seja feita pela rede, ou se admita que a rede pode modificar a marcação do tráfego, será possível utilizar uma política activa para analisar a carga da rede, e seleccionar o melhor mapeamento a realizar em cada momento. Assim, em situações de sobrecarga, em que não é possível assegurar a classe de tráfego pretendida por todos os utilizadores, será possível fazer uma selecção mais restritiva das ligações que podem utilizar as classes de tráfego mais prioritárias, melhorando a disponibilidade para as ligações mais importantes, e degradando a classe de tráfego para as ligações menos importantes. É também possível que o operador da rede possa remarcar ou descartar tráfego no caso de problemas internos da rede, como seja a falha de linhas que reduza significativamente a capacidade de transporte de tráfego da rede.

Neste nível, a gestão da rede faz a monitorização dos atrasos, *jitter*, débito e perda de pacotes que cada cliente tem, como forma de assegurar o cumprimento do

SLA. É ainda feita a contabilização dos custos a facturar a cada cliente e monitorizada a disponibilidade da rede e dos serviços que ela oferece, como forma de assegurar o cumprimento do CLA.

### **5.2.3 Nível Rede e Sub-Rede**

Neste nível são geridos o estabelecimento de linhas extra, a largura de banda das linhas existentes, e a admissão de novas ligações. Para simplificar a implementação, dividiram-se estas funcionalidades por dois níveis: o nível sub-rede é responsável por uma região da rede, e o nível rede propriamente dito é responsável por toda a rede, ou seja, o conjunto de todas as sub-redes.

#### **5.2.3.1 Gestão da Admissão de Novas Ligações**

Tal como se referiu na secção 5.1.2, para assegurar a qualidade de serviço torna-se necessário implementar um mecanismo de controlo de admissão de novas ligações. Uma das formas de o fazer é através de corretores de largura de banda. Para suportar o seu funcionamento é necessário saber qual a capacidade que pode ser disponibilizada para ser reservada pelos utilizadores. Para isso é necessário introduzir alguns conceitos de teoria de grafos.

Um corte X-Y numa rede é o conjunto de linhas que, quando retiradas, desligam o nó X do nó Y. O corte mínimo é o corte de menor capacidade. O teorema Max-Flow Min-Cut [Tanenbaum 81a] diz que o fluxo máximo de dados entre dois nós arbitrários numa rede é igual à capacidade do corte mínimo que separa esses dois nós. A figura 5.7 ilustra uma rede onde todas as linhas são de 10 Mbps. Está assinalado um dos cortes mínimos entre os nós 1 e 6, que corresponde a um fluxo máximo entre estes dois nós de 20 Mbps.

Um problema que fica logo patente na figura 5.7 é que para atingir este fluxo máximo é necessário saturar as linhas do corte mínimo, o que significa também que outros nós que pretendam comunicar utilizando alguma das linhas desses caminhos entre 1 e 6 não encontrarão qualquer capacidade disponível nas linhas. Além disso, para se conseguir saturar o corte mínimo apenas com o tráfego entre 1 e 6, é necessário utilizar um protocolo de encaminhamento que divida o tráfego pelos vários caminhos existentes. Na maior parte das redes, normalmente apenas é utilizado um único caminho

em cada instante. Um outro problema de saturar linhas é que os atrasos sofridos pelos pacotes nas suas filas de espera tenderão para infinito.

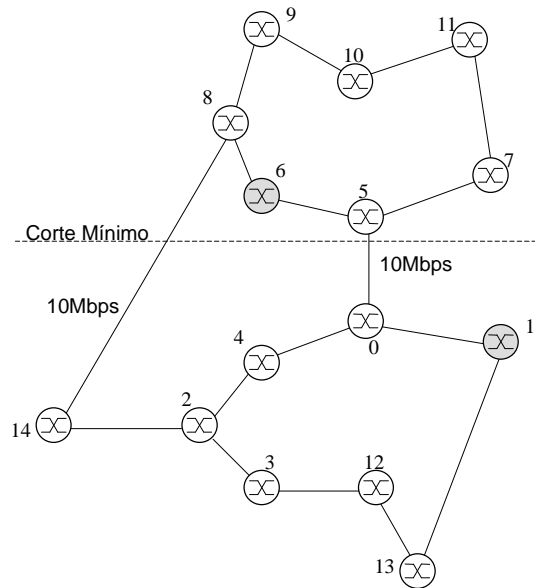


Figura 5.7: Exemplo de fluxo máximo no corte mínimo de uma rede.

Caso se admita que o protocolo de encaminhamento pode escolher e modificar as rotas independentemente do corretor de largura de banda, então o corretor de largura de banda só pode aceitar reservas até ao valor do menor dos cortes mínimos existente na rede. Para a rede deste exemplo, só devia aceitar reservas até 20 Mbps. Caso se admita que o protocolo de encaminhamento só usa um único caminho entre quaisquer dois nós, então o corretor de largura de banda só pode aceitar reservas até perfazer a capacidade da linha de menor capacidade da rede. Para a rede deste exemplo, só devia aceitar reservas até 10 Mbps. Assim, se por exemplo, os clientes fizessem todos reservas de 64 Kbps, seria possível ter  $10000/64 = 156$  clientes em simultâneo. Estes valores terão ainda de ser divididos pelas várias classes de tráfego, e limitados consoante o atraso máximo admitido para as filas de espera, para que este não tenda para infinito. A figura 5.8 ilustra a variação do tempo médio que um pacote demora num sistema com uma fila de espera de dimensão 60, em termos do tempo de transmissão médio de um pacote, como função da intensidade de tráfego  $\rho$ , admitindo-se que os processos de partidas e

chegadas são de Poisson. Este caso corresponde a um sistema M/M/1/60<sup>2</sup> [Keiser 89a]. Desta figura conclui-se facilmente que à medida que a intensidade do tráfego se aproxima de 1, correspondendo à taxa de chegadas de pacotes se aproximar da taxa de partidas, os tempos de atraso que os pacotes sofrem para serem transmitidos crescem exponencialmente, tendendo-se para a situação em que a fila de espera transborda, passando-se assim para uma situação de descarte de pacotes.

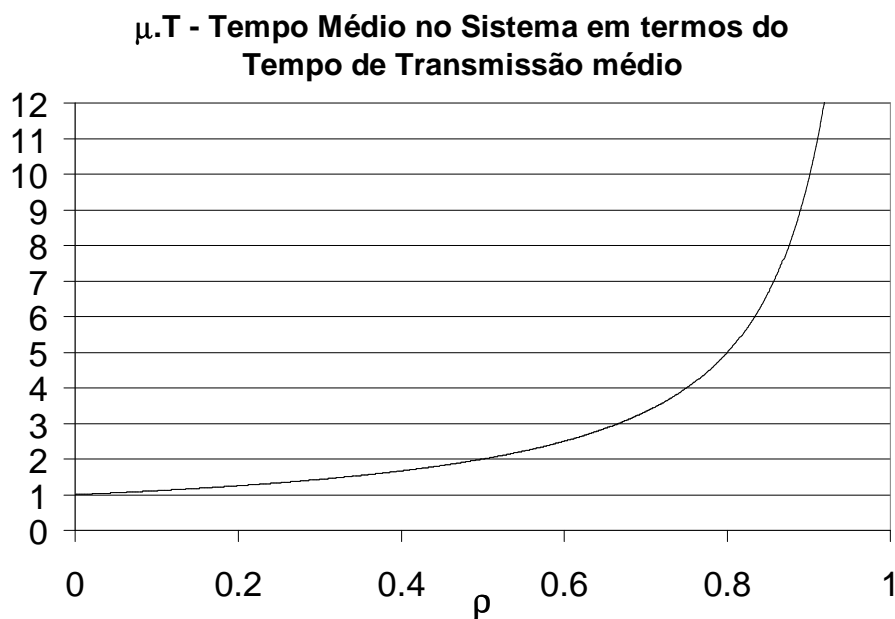


Figura 5.8: Variação do tempo médio que um pacote demora num sistema de fila de espera em função da intensidade de tráfego  $\rho$ .

Uma vez que a situação descrita corresponde ao pior caso possível de saturação do corte mínimo, será natural que, na rede como um todo, seja possível transportar mais tráfego em situações normais. Na realidade, é fácil pensar em vários pares de nós que comuniquem através de percursos totalmente disjuntos e que por isso poderiam utilizar toda a largura de banda disponível. Também em muitos casos o tráfego não é contínuo, mas por rajadas, ou principalmente numa única direcção, pelo que será possível

---

<sup>2</sup> Esta notação, devida a Kendall, é na sua forma mais geral dada por A/B/c/K/N/Z, onde os símbolos designam: A - distribuição dos tempos entre chegadas consecutivas; B - disciplina de serviço; c - número de servidores; K - capacidade do sistema (tamanho da fila de espera, ou dimensão dos *buffers*); N - número de potenciais clientes na população; Z - disciplina da fila de espera. Os símbolos mais usados para A e B são: M - Markov (normalmente exponencial); D - determinístico (normalmente constante); G - geral; GI - geral independente.

comportar um maior número de clientes. Além disso, é possível utilizar numa classe de tráfego largura de banda que sobre de outras classes de tráfego, o que também permite aumentar o número de clientes.

Pretende-se que a classe de envio expedito (Ouro) funcione como um circuito virtual, oferecendo garantias rígidas de qualidade. Assim, para esta classe, o corretor de largura de banda deve trabalhar sempre prevendo o pior caso possível. Quando não houver mais largura de banda disponível no corretor, tem de se recusar os pedidos de largura de banda, a menos que se coloque mais largura de banda nas linhas, ou mais linhas na rede. Isto pode ser feito com uma política activa que faça a gestão de pedidos de mais largura de banda, ou a gestão do estabelecimento de ligações extra quando e onde for necessário e possível, como será descrito na subsecção seguinte (5.2.3.2).

Já para a classe de envio assegurado (Prata) admite-se alguma perda de pacotes. Assim, será possível flexibilizar um pouco a entrada de novos clientes, caso a carga nas linhas não seja excessiva. Isto pode ser implementado com uma política activa que monitoriza a carga nas linhas e os descartes de pacotes, para indicar se podem, ou não, ser admitidos novos clientes.

Finalmente, para a classe melhor esforço não é feito nenhum controlo sobre os clientes, ou a qualidade de serviço, sendo todas as ligações aceites, tal como acontece na Internet actual, provocando uma degradação do serviço tanto maior quanto mais intensamente se utiliza a rede.

### **5.2.3.2 Gestão da Largura de Banda de Linhas**

Se pensarmos que o fornecedor de serviço Internet transporta o tráfego Internet sobre linhas de grande capacidade (normalmente fibras ópticas), que podem ser partilhadas com outros serviços, tais como a rede telefónica, Frame Relay, ATM, ou mesmo X.25, será possível pensar que a largura de banda instalada possa ser desviada dinamicamente de uns serviços para outros consoante a procura em cada instante. Este desvio pode ser feito de forma fixa consoante a hora do dia, ou dia da semana. Alternativamente, podem-se utilizar políticas activas que ajustem dinamicamente a largura de banda atribuída a cada serviço consoante as necessidades de cada instante. Trata-se, neste caso, da utilização de um sistema do tipo largura de banda a pedido (*Bandwidth on Demand*, ou BOD). Um exemplo de um sistema deste tipo é descrito em [RFC 1990] para operar sobre a Rede Digital com Integração de Serviços (RDIS). Os

cenários de redes sem fios [Toh 02] são outro caso onde também é normal poder ajustar-se o ritmo de transmissão entre certos limites.

A perspectiva nesta tese não é o uso sistemático da facilidade de modificação de largura de banda para resolução dos problemas, mas sim ilustrar como se poderia utilizar esta facilidade se estivesse disponível. Seria natural que o uso frequente desta facilidade para aumentar a largura de banda conduzisse, fruto de actividades de planeamento, a um aumento real da capacidade da rede.

Admitindo que a largura de banda disponível pode ser ajustada às necessidades dentro de certos limites, é possível que a gestão da admissão de novas ligações tome em consideração este facto. Assim, minimizam-se as recusas de estabelecimento de ligação aos utilizadores do serviço, maximiza-se a qualidade de serviço oferecida aos utilizadores e simultaneamente maximiza-se o lucro obtido pelo fornecedor de serviço. Será ainda possível ajustar a largura de banda das linhas de forma a manter a carga num nível que permita obter o atraso médio pretendido para os pacotes, de acordo com a figura 5.8.

Seria ainda possível pensar que o fornecedor de serviço Internet poderia alugar banda temporariamente de outro operador se o lucro obtido pelo fornecimento do serviço associado à utilização dessa largura de banda fosse suficiente para a pagar, ou se o prejuízo decorrente de esse serviço não ser oferecido fosse superior ao preço dessa largura de banda. Neste trabalho não se tentou abordar o problema dos preços a pagar pelos utilizadores, que poderiam depender da situação da rede na altura da utilização do serviço. Em [DaSilva 00][Falkner 00] são dadas perspectivas dos métodos de determinação de preços em redes.

### **5.2.3.3 Gestão de Falhas de Linhas**

Caso haja falha de alguma linha, a capacidade de transporte de tráfego da rede é reduzida, podendo mesmo perder-se entre alguns nós da rede, resultando numa falha de disponibilidade do serviço. Para tentar minimizar este problema, alguns *routers* têm a possibilidade de estabelecer linhas de recurso para outros *routers* com essa mesma funcionalidade.

Sempre que há uma falha de uma linha, é activada uma política activa que executa um algoritmo de pesquisa de qual a melhor ligação de recurso a estabelecer. Este algoritmo é composto por um algoritmo responsável por cada região da rede



independentemente das outras e por um algoritmo de topo gerindo as ligações entre regiões. Os dois tipos de algoritmo começam nos nós, ou regiões, onde ocorreu a falha, e procuram, de entre as possíveis linhas de recurso, qual a melhor que resolve o problema. Se a falha tiver sido interna a uma região, e o algoritmo dessa região tiver conseguido resolver o problema, nada mais é feito. No caso contrário, o algoritmo de topo tem de intervir. Em qualquer dos casos, o gestor da rede é notificado para que a falha seja resolvida o mais rapidamente possível.

#### **5.2.4 Nível Equipamento**

O nível equipamento funciona localmente a cada equipamento de rede, gerindo os seus parâmetros de configuração para resolver problemas locais e otimizar o desempenho.

Seguidamente são descritos os parâmetros de configuração de um *router* com serviços diferenciados e os compromissos patentes na escolha dos valores de cada um, focando-se os problemas de gestão da largura de banda das linhas e gestão do descarte de pacotes.

##### **5.2.4.1 Gestão da Largura de Banda por Classe de Tráfego**

Para suportar serviços diferenciados, pretende-se que seja possível gerir a largura de banda das linhas de saída que é utilizada por cada classe de tráfego. Pretende-se também garantir uma certa largura de banda mínima por classe de tráfego, devendo ainda a largura de banda não utilizada por uma classe de tráfego ser distribuída pelas restantes classes de tráfego proporcionalmente às suas importâncias.

Existem vários mecanismos para dividir a largura de banda entre as diferentes classes de tráfego [Kilki 99a]. O mecanismo de serviço das filas de espera baseado em prioridades em sentido estrito, como o *Priority Queueing*, tem o problema de as filas de espera mais prioritárias poderem esgotar toda a largura de banda disponível, não deixando nada para as menos prioritárias. São preferíveis os métodos que controlam o serviço das filas com base em pesos, pois asseguram as condições de haver uma largura de banda mínima por classe de tráfego e redistribuir a largura de banda que sobra pelas outras classes de tráfego.

Os mecanismos derivados de *Fair Queueing* [Peterson 96a] pretendem aproximar o resultado que se obteria tendo uma disciplina de serviço de *Round Robin* ao bit. Isto é,

calculam qual seria o instante de transmissão de cada pacote, se fosse transmitido rotativamente um bit de dados de cada fila de espera, sendo o próximo pacote a ser transmitido o que devia acabar de transmitir em primeiro lugar, de acordo com este cálculo. Na versão com pesos, os pesos indicam o número de bits a servir de cada fila em cada rodada. Estes mecanismos exigem um processamento por fluxo de dados relativamente complexo, mas são os que dão mais garantias de justiça entre as várias fontes.

Um mecanismo mais simples de implementar é o *Deficit Round Robin* [Keshav 97a], onde as filas de espera são percorridas rotativamente, sendo atribuído a cada uma um certo número de créditos de transmissão. Caso uma fila fique com créditos suficientes para transmitir o seu primeiro pacote, este é transmitido, e os créditos são diminuídos pela dimensão do pacote transmitido. Caso uma fila não tenha pacotes para transmitir, não lhe são atribuídos créditos para evitar a sua acumulação. Na variante com pesos, o número de créditos atribuído a cada fila, em cada rodada, é proporcional ao respectivo peso. Embora simples de implementar, este mecanismo é injusto em escalas de tempo inferiores ao tempo de uma rodada, que pode ser significativo se houver pacotes grandes.

Optou-se por utilizar um mecanismo de serviço baseado em *Weighted Deficit Round Robin* com uma fila de espera para cada classe DiffServ. Este mecanismo é parametrizado pelos pesos de cada classe de tráfego, e pelo número de créditos distribuídos em cada rodada. Na tabela 5.1 ilustra-se um exemplo dos pesos com apenas três classes de tráfego, e as correspondentes fracções da largura de banda da linha de saída mínimas para cada classe de tráfego.

| Classe | Peso | Fracção da Largura de Banda |
|--------|------|-----------------------------|
| EF     | 5    | 5/10                        |
| AF     | 4    | 4/10                        |
| BE     | 1    | 1/10                        |
| total  | 10   | 10/10                       |

Tabela 5.1: Exemplo de distribuição da largura de banda pelas classes de tráfego.

Como se pretende que nunca falem recursos para as classes EF e AF, estas ficam com a maior parte da largura de banda reservada. Utilizou-se um peso de 1 para a

classe BE de forma a garantir-lhe um mínimo de largura de banda. Desta forma, o tráfego da classe BE que exceder este mínimo apenas pode utilizar a largura de banda que sobra das outras classes. Esta sobra é distribuída na proporção dos pesos das classes. Por exemplo, se o tráfego AF for zero, EF fica com 5/6 e BE com 1/6 da largura de banda.

Esta divisão levanta um problema: como saber as fracções adequadas em cada momento para cada linha da rede? Certamente que usar os mesmos valores em todas as linhas para todo o tempo pode causar problemas pontuais. Na realidade, pode acontecer que numa certa linha haja durante um certo período um excesso de tráfego numa dada classe, que pode conduzir a que essa classe de tráfego obtenha uma qualidade de serviço pior que outras classes menos prioritárias, se nada for feito. Esta situação pode facilmente ocorrer caso se utilizem as quatro classes de tráfego assegurado (AF1-AF4) previstas em [RFC 2597], e se pretenda que haja uma ordenação em termos de qualidade de serviço. Para resolver o problema pode-se aumentar a largura de banda disponibilizada para a classe cuja qualidade de serviço se degradou, diminuindo a das classes menos importantes. Isto pode ser feito sabendo os percursos seguidos por cada fluxo de dados, e calculando os pesos das filas adequados para cada nó, o que contraria os princípios de DiffServ, em que se pretende processar apenas tráfego agregado no núcleo da rede. Alternativamente, pode-se recorrer a políticas activas de gestão que monitorizem cada fila de espera, e ajustem dinamicamente os pesos, se tal for necessário. Esta solução tem a vantagem de as políticas activas serem locais a cada nó.

#### **5.2.4.2 Gestão do Descarte de Pacotes**

Quando se atinge uma situação de congestão é necessário descartar pacotes. O mecanismo mais simples consiste em descartar pacotes quando a fila de espera transborda. Este mecanismo pode ser melhorado definindo limiares de descarte diferentes consoante a prioridade dos pacotes. Desta forma, os pacotes de menor prioridade (fora do perfil negociado, ou das classes menos importantes) serão descartados primeiro.

Um outro problema advém da utilização do protocolo TCP, que aumenta o ritmo de transmissão até haver descarte de pacotes, caso em que diminui a janela de transmissão, reduzindo o ritmo. No pior caso, todas as fontes aumentam progressivamente o ritmo de transmissão, pelo que a fila de espera acaba por

transbordar, resultando no descarte de pacotes para quase todas as fontes, e a consequente redução do ritmo simultaneamente nas várias fontes. Isto pode causar um efeito de sincronização global que faz com que a carga oscile entre as situações de congestão num extremo, e uma situação de carga reduzida no outro extremo.

O mecanismo de Detecção Antecipada Aleatória (*Random Early Detection*, ou **RED** [Floyd 93]) foi desenvolvido para operar com o protocolo TCP para detectar antecipadamente situações de congestão, e tentar preveni-las. Este mecanismo funciona começando a descartar alguns pacotes antes de a fila de espera transbordar, na esperança que o protocolo TCP, ao detectar perda de pacotes, reduza o ritmo de transmissão, evitando que a fila de espera realmente transborde, e prevenindo assim a congestão. Adicionalmente, pretende-se limitar o comprimento médio da fila de espera de forma a reduzir os atrasos, e ainda evitar o problema de sincronização global.

O mecanismo RED ajusta a probabilidade de descarte de pacotes consoante o valor actual do tamanho médio da fila de espera conforme ilustrado na figura 5.9. Caso o tamanho médio da fila de espera se encontre abaixo de um limiar mínimo, a probabilidade de descarte é nula. À medida que ele cresce entre o limiar mínimo e máximo, também a probabilidade de descarte vai crescendo linearmente até um valor  $P_{máx}$ , tornando-se 1 para valores do tamanho médio da fila de espera superiores ao limiar máximo.

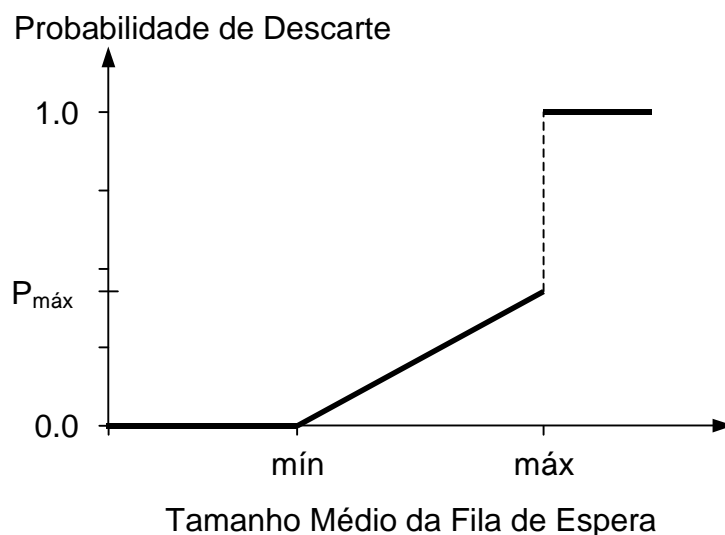


Figura 5.9: Probabilidade de descarte de pacotes no mecanismo de Detecção Antecipada Aleatória (RED).

Com o objectivo de permitir algumas rajadas de dados, o tamanho médio da fila de espera é calculado utilizando um filtro passa-baixo com uma média móvel de peso exponencial (EWMA), dada pela equação 5.1:

$$m\u00e9dia \leftarrow (1 - w_q) \cdot m\u00e9dia + w_q \cdot q \quad (5.1)$$

O peso  $w_q$  corresponde ao peso do comprimento actual da fila de espera ( $q$ ) na média. Se  $w_q$  for muito baixo, a média responde muito devagar a variações no comprimento da fila de espera. Se  $w_q$  for muito elevado, haverá descarte de pacotes quando houver uma rajada de dados.

Na figura 5.10 ilustra-se o funcionamento do filtro passa-baixo. Esta figura corresponde a um exemplo simulado de funcionamento de uma fila de espera com tráfego AF durante 20 segundos gerado por fontes TCP. O comprimento da fila é de 120 pacotes, o peso para calcular o comprimento médio é  $w_q=0.01$ . Os limiars do RED são  $m\u00edn=30$ ,  $m\u00e1x=60$ ,  $P_{m\u00e1x}=1/50$ .

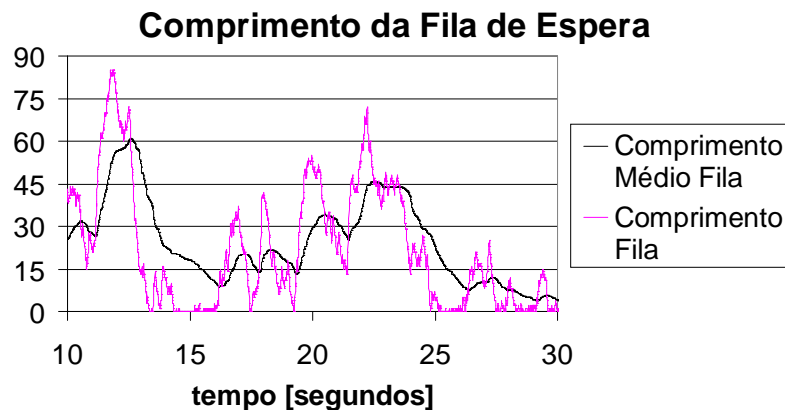


Figura 5.10: Variação do tamanho médio da fila de espera.

Da análise desta figura, verifica-se que o comprimento médio da fila de espera segue as variações do comprimento da fila, filtrando as variações bruscas, permitindo assim algumas rajadas de dados, tal como se pretendia. Nota-se ainda o efeito de o comprimento médio da fila ultrapassar o limiar de 60 pacotes, em que todo o tráfego que chega à fila de espera passa a ser descartado, pelo que o comprimento da fila é rapidamente reduzido. Quando o comprimento médio da fila ultrapassa o limiar de 30 pacotes, começa a haver algum descarte de pacotes, com probabilidade crescente. À medida que começa a haver descarte de pacotes, as fontes TCP onde houve descarte

reagem reduzindo o ritmo de transmissão, pelo que o efeito de o comprimento médio da fila ultrapassar o limiar de 30 pacotes causa uma redução do comprimento da fila, pouco perceptível na figura.

Desta análise podem-se tirar algumas conclusões para auxiliar o dimensionamento dos diversos parâmetros. O limiar máximo define um limite para o comprimento médio da fila, que deve ser escolhido de acordo com o atraso máximo pretendido para os pacotes. Quanto maior for o tamanho médio da fila de espera, maiores são os atrasos sofridos pelos pacotes.

A dimensão da fila de espera deve ser suficiente para que não transborde sem o comprimento médio atingir primeiro o limiar máximo, por forma a que o mecanismo RED tenha um funcionamento efectivo.

A equação 5.2 [Floyd 93] permite obter um limite para o peso  $w_q$  de forma a permitir rajadas<sup>3</sup> de dimensão  $L$  pacotes sem descartes:

$$L + 1 + \frac{(1 - w_q)^{L+1} - 1}{w_q} < \text{mín} \quad (5.2)$$

Se, por exemplo, o limiar mínimo for  $\text{mín}=30$  e o tamanho máximo da rajada for  $L=50$ , será necessário escolher  $w_q \leq 0.043$ .

Como regra prática, o limiar máximo deve ser pelo menos o dobro do limiar mínimo [Floyd 93]. Para o comprimento máximo da fila de espera pode-se usar como regra prática o valor de duas vezes o limiar máximo.

Uma vez que numa classe de serviço assegurado de DiffServ estão previstos três níveis de probabilidade de descarte, será lógico utilizar três conjuntos de parâmetros para o RED, a aplicar ao tráfego de cada um dos diferentes níveis de probabilidade de descarte [RFC 2698]. Neste caso, o comprimento médio utilizado pode ser calculado de diferentes formas [Goyal 99]. Pode ter-se apenas uma única média, calculada pelo número total de pacotes, independentemente das suas marcações. Uma segunda forma é ter tantas médias quantos os níveis de probabilidade de descarte, calculando-se cada uma com os pacotes da respectiva probabilidade de descarte e das probabilidades de descarte mais baixas. Uma terceira forma é continuar a ter tantas médias quantos os

---

<sup>3</sup> Assume-se que a fila de espera está inicialmente vazia, com um comprimento médio nulo, e que a fila cresce de 0 a  $L$  pacotes, correspondendo a  $L$  chegadas de pacotes e nenhuma partida.

níveis de probabilidade de descarte, mas utilizar em cada média apenas os pacotes da sua probabilidade de descarte.

A figura 5.11 ilustra os três conjuntos de limiares para a segunda forma de calcular a média. Os limiares mais elevados são aplicados ao comprimento médio da fila de espera calculado apenas com os pacotes do nível mais prioritário (AFx1), controlando o respectivo descarte de pacotes. Os limiares seguintes são aplicados ao comprimento médio calculado com o conjunto dos pacotes dos dois níveis mais prioritários, e controlam o descarte dos pacotes do nível de probabilidade de descarte intermédio (AFx2). Os limiares mais baixos são aplicados ao comprimento médio calculado com o conjunto de todos os pacotes independentemente do nível de probabilidade de descarte, e controlam o descarte dos pacotes do nível de probabilidade de descarte mais baixo (AFx3).

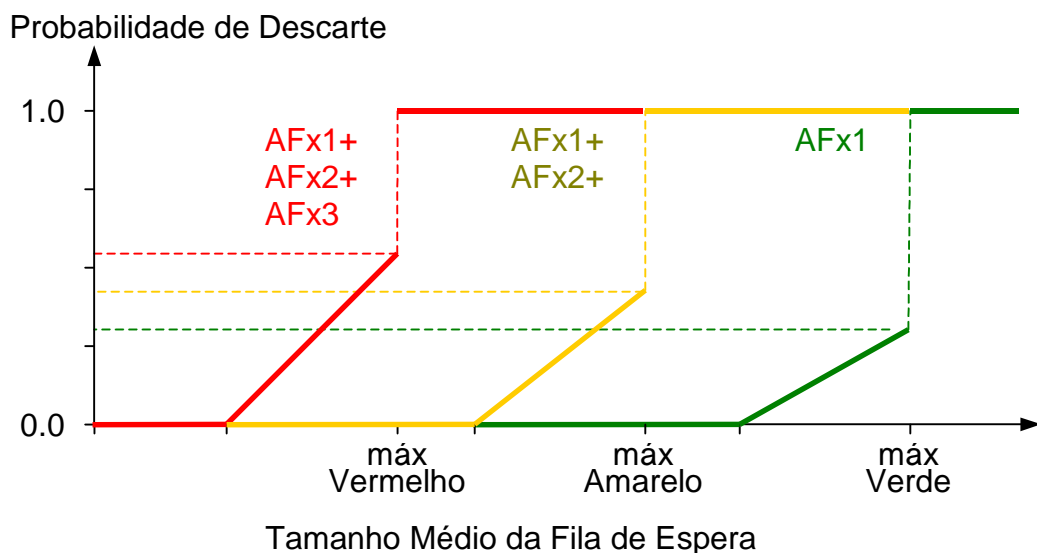


Figura 5.11: Limiares RED para uma marcação com três cores.

O descarte de pacotes para os níveis de probabilidade de descarte maior é mais agressivo de três maneiras. Em primeiro lugar, o limiar mínimo é menor, começando-se a descartar pacotes mais cedo. Em segundo lugar, a probabilidade de descarte  $P_{máx}$  é maior, descartando-se mais pacotes. Finalmente em terceiro lugar, o limiar máximo é menor, descartando-se todo o tráfego de menor prioridade antes do de maior prioridade.

A configuração destes parâmetros permite controlar quanto tráfego fora de perfil é que se deixa passar, podendo-se recorrer a políticas activas de gestão para ajustar dinamicamente a quantidade de tráfego fora de perfil que se deixa passar.

Uma versão simplificada deste mecanismo de descarte de pacotes corresponde a utilizar apenas dois conjuntos de parâmetros a aplicar ao tráfego dentro e fora de perfil, chamando-se o mecanismo **RIO: RED** com bit *in/out*, pois neste caso basta um bit para distinguir a marcação dentro de perfil da fora de perfil.

Para concluir esta secção, deve-se ainda referir que na classe de serviço melhor esforço utiliza-se apenas um conjunto de parâmetros para o RED visto que não há definição de perfil. Na classe de serviço expedito pretende-se simular um circuito virtual, pelo que é preferível não utilizar o mecanismo RED, havendo descarte apenas quando a fila transborda, o que não deve acontecer se as reservas forem bem feitas e respeitadas.

## 5.3 SIMULAÇÃO

As políticas activas foram testadas numa rede simulada. Nesta secção descreve-se o simulador, a rede simulada, os parâmetros de simulação utilizados e, por fim, discutem-se as opções tomadas.

### 5.3.1 O *Network Simulator*

Para simular uma rede para testar as políticas activas utilizou-se o simulador de redes do projecto VINT (*Virtual InterNetwork Testbed*): o *Network Simulator*, ou *ns* [NS], versão 2.1b5, com uma versão modificada das adições ao *ns* para serviços diferenciados de [NS-DiffServ]. Foram ainda introduzidas modificações adicionais no simulador com o objectivo de recolher estatísticas extremo a extremo por fonte de tráfego, e também estatísticas em cada linha por classe de tráfego.

O simulador de redes *ns* tem um motor de simulação extensível, implementado em C++, que utiliza a linguagem de comandos do MIT OTcl [Wetherall 95], (uma versão orientada para objectos de Tcl, a Tool Command Language [Ousterhout 94]), para configuração e programação de comandos. O *ns* é um simulador controlado por eventos. Os eventos são escalonados no *ns* permitindo invocar procedimentos OTcl em instantes arbitrários de tempo de simulação. Estes procedimentos oferecem um mecanismo de simulação flexível, podendo ser utilizados para iniciar ou terminar fontes de tráfego, recolher estatísticas, modificar parâmetros de simulação, etc.



O *ns* permite simular o funcionamento interno das redes incluindo vários tipos de filas de espera, com todo o processamento de pacotes associado, redes locais, sem fios, ou de satélite, vários protocolos de encaminhamento com ou sem *multicast*, e ainda o funcionamento de um vasto conjunto de protocolos de transporte e de nível aplicação, destacando-se: UDP, TCP, RTP (*Real-Time Protocol*), Telnet, FTP (*File Transfer Protocol*), HTTP, fontes CBR (*Constant Bit Rate*), e Poisson.

Com o *ns* está disponível um programa animador do funcionamento da rede, o *Network Animator*, ou *nam*, que permite observar os pacotes a fluir na rede, desde que se configure o simulador para gerar um ficheiro com a informação necessária à animação. Este programa é útil para depuração de erros e, de um ponto de vista didáctico, para melhor compreender o funcionamento da rede e dos protocolos utilizados.

Em termos de protocolo de encaminhamento, estão disponíveis três alternativas no *ns*:

- Encaminhamento estático, que é calculado uma vez no início da simulação pelo algoritmo de caminho mais curto de Dijkstra e depois nunca muda.
- Encaminhamento sessão, calculado pelo algoritmo de caminho mais curto de Dijkstra automática e instantaneamente sempre que há uma modificação no estado de alguma linha.
- Encaminhamento por vector de distâncias, simula o algoritmo de encaminhamento por vector de distâncias com separação de horizontes (*split horizon*) e envenenamento inverso (*poisoned reverse*).

Infelizmente, a nível dos protocolos de encaminhamento disponíveis, verificou-se que a única métrica disponível era o custo dos caminhos, não estando disponíveis métricas alternativas, que podia ser interessante utilizar, tais como o atraso, a largura de banda, ou a fiabilidade. Uma vez que o algoritmo de encaminhamento por vector de distâncias tem problemas de convergência para infinito no caso de falha de linhas, que podem tornar as simulações extraordinariamente lentas, visto que são simulados todos os pacotes do algoritmo de encaminhamento, optou-se por utilizar o encaminhamento sessão.

Embora o *Ptolemy*, utilizado no capítulo anterior, seja um bom simulador, facilmente extensível e com uma extensa biblioteca de componentes que simulam vários tipos de equipamentos, especialmente no domínio do processamento de sinais, tem

relativamente poucos componentes disponíveis para simular redes e protocolos. Desta forma, optou-se por usar outro simulador mais adequado à simulação de redes e protocolos.

Escolheu-se utilizar o simulador *ns* por ser disponibilizado gratuitamente com o código fonte, pela sua versatilidade, pelo realismo dos protocolos simulados, e por haver disponível uma implementação simplificada de serviços diferenciados. Esta implementação apenas suportava dois níveis de prioridade de descarte em apenas uma classe AF para marcação a duas cores. Esta implementação foi estendida para suportar as quatro classes AF propostas em [RFC 2597], mantendo-se os dois níveis de prioridade de descarte controlados por um mecanismo RIO, que se considerou suficiente. Foram ainda feitas modificações ao mecanismo de serviço das filas de espera *Weighted Deficit Round Robin* de [NS-DiffServ] para corrigir dois pequenos problemas existentes, limitando o número de créditos das filas que não têm pacotes a transmitir, para que não cresçam indefinidamente, e continuando a fazer rodadas de distribuição de créditos enquanto houver algum pacote a transmitir, para evitar que um pacote grande fique por transmitir por não ter créditos suficientes. Esta última modificação faz com que a linha de transmissão nunca fique livre se houver pacotes a transmitir, tornando o mecanismo de serviço conservador de trabalho (*work-conserving*), se bem que possa contribuir para um aumento do *jitter*.

Em complemento ao trabalho desta tese foi desenvolvido um editor gráfico com interesse didáctico [Pereira 01b], que facilita o trabalho de geração de topologias de rede, sua configuração e teste, pois produz o código OTcl necessário para o *ns*, corre a simulação e apresenta os resultados graficamente.

### 5.3.2 Topologia da Rede

A figura 5.12 ilustra a topologia do núcleo da rede simulada. O núcleo da rede contém 15 nós divididos por 3 regiões. Todas as linhas têm uma capacidade de 10 Mbps e um atraso, para simular o tempo de processamento dos pacotes e o tempo de propagação na linha, de 5 ms. Os *routers* de cor cinzenta têm a possibilidade de activar linhas extra entre eles, com a mesma capacidade e atraso das restantes linhas, para utilizar em situações de falhas ou de sobrecarga de linhas.

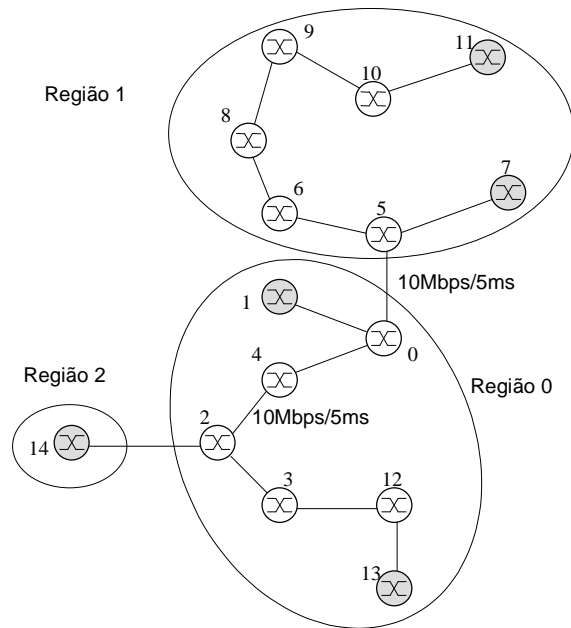


Figura 5.12: Topologia do núcleo da rede simulada.

Para simular a geração de tráfego, admitiu-se que havia um conjunto de pontos de acesso (PoPs) que geravam tráfego correspondente ao funcionamento normal da Internet.

A figura 5.13 ilustra como as fontes de tráfego são ligadas ao núcleo da rede através de um PoP. Cada PoP está ligado a um nó do núcleo por uma linha de 2 Mbps com um atraso de 1 ms. Se simplesmente se ligassem as fontes de tráfego ao nó do PoP, não haveria qualquer limitação ao tráfego que uma fonte particular poderia injectar na rede, o que não corresponde a uma situação real. Assim, se quisermos, por exemplo, simular um conjunto de modems a gerar tráfego para a rede, devemos ligar cada fonte de tráfego ao nó do PoP por uma linha com capacidade igual ao ritmo de transmissão do modem. Se quisermos simular uma rede empresarial, devemos ligar as fontes de tráfego ao nó do PoP por linhas a 10 Mbps para simular uma rede local, e colocar em cada fonte que gera tráfego para a rede um condicionador para limitar o tráfego nas classes EF e AF.

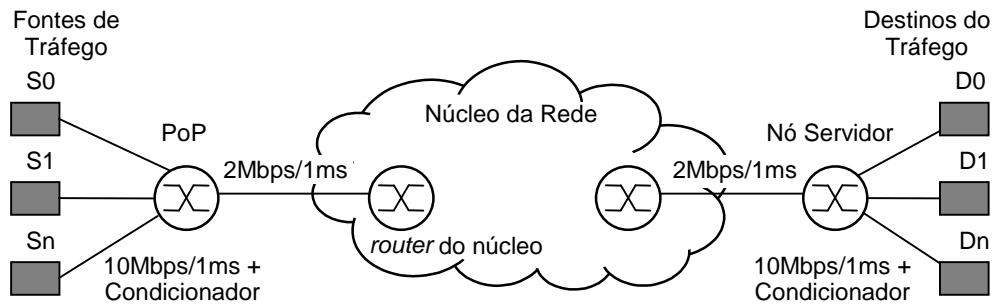


Figura 5.13: Modelo de fontes de tráfego ligadas a um *router* do núcleo.

Embora na figura 5.13 se tenha separado as fontes de tráfego e os destinos em nós separados para simplificar a explicação, na realidade os nós dos PoPs servem simultaneamente como nós de passagem para o tráfego gerado e recebido. Isto é, os nós que servem de fonte de tráfego também funcionam como destino para outros PoPs da rede.

A figura 5.14 ilustra a rede no *nam*. Os nós 0 a 14 são os nós do núcleo da rede. Os nós 15 a 29 são os PoPs, e os nós 30 a 44 contêm os agentes, não visíveis no *nam*, que simulam as aplicações que geram e recebem tráfego da rede. O animador permite ver todos os pacotes, identificados por cores diferentes consoante a classe de tráfego. Permite ver as filas de espera e o descarte de pacotes: na linha do nó 20 para o 5 está a haver descarte de pacotes AF. Permite, também, seguir o percurso de pacotes com monitores: está ilustrado um monitor como um 1 para um pacote que vai da linha do nó 5 para o 7. E permite ainda ver a carga de uma linha ao longo da simulação: está ilustrada a carga da linha 0-5, nas duas direcções, nas janelas por baixo do indicador do tempo de simulação.

### 5.3.3 Parâmetros de Simulação

Para simular a geração de tráfego num PoP, admitiu-se que chegavam novas ligações a intervalos dados por uma distribuição exponencial com uma dada média, e que cada ligação tinha uma duração dada por uma distribuição exponencial com uma dada média.

O tráfego oferecido à rede é dado pelo produto da taxa de chegada de novas ligações pela sua duração média.

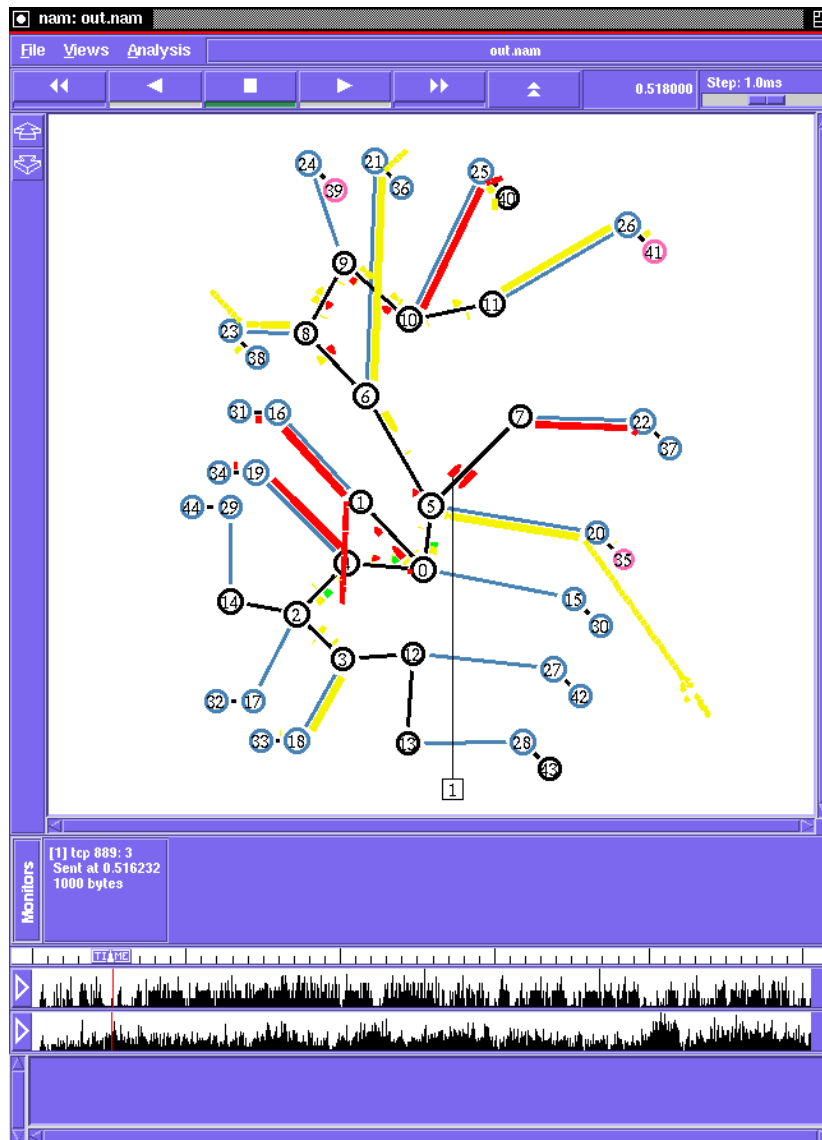


Figura 5.14: Rede no *Network Animator*.

Nas situações em que se pretendia comparar várias alternativas de gestão, utilizou-se a mesma semente para os geradores de números aleatórios, de maneira a gerar exactamente o mesmo padrão de ligações, tornando, assim, a comparação válida.

Admitiu-se que em cada PoP havia um número máximo de ligações simultâneas, e que as ligações em excesso eram recusadas. Este caso corresponde a um processo M/M/s/0 [Keiser 89b], que pretende simular um sistema com recursos limitados, como acontece na realidade.

Para minimizar o efeito do transitório inicial sobre os resultados das simulações, configuram-se as simulações para começar com uma carga inicial de ligações correspondente ao tráfego oferecido pretendido. Depois de começar a simulação,

admitiu-se que chegavam em média 6 novas ligações por segundo, ajustando-se a sua duração média consoante o valor pretendido para o tráfego oferecido à rede.

Admitiu-se que cada ligação correspondia ao tráfego de apenas uma das seguintes aplicações, sendo todas igualmente prováveis:

- CBR/UDP: gera tráfego a um ritmo constante.
- Telnet/TCP: gera pacotes com intervalos tirados da distribuição telnet da tcplib.
- OnOff/UDP: gera tráfego em rajadas de ritmo constante durante períodos On, alternados por períodos Off sem tráfego. A duração das rajadas e os intervalos entre rajadas são tirados de distribuições exponenciais com a mesma média (1 segundo), de forma a que se esteja a transmitir em média durante 50% do tempo.
- HTTP/TCP: gera tráfego simulando a transferência de páginas WWW. As páginas são todas compostas de um objecto principal de 1 Kb e dez objectos de 6 Kb. Os intervalos entre pedidos consecutivos são tirados de uma distribuição exponencial de média 25 ms.
- FTP/TCP: gera tráfego ininterrupto para transmissão por TCP.

Admitiu-se que o destino de cada ligação era escolhido aleatoriamente de entre os outros PoPs, sendo todos igualmente prováveis.

Admitiu-se que o tráfego oferecido à rede era igual em todos os PoPs.

Admitiu-se que o tráfego gerado, em todos os PoPs, era em 1/4 dos casos de saída e em 3/4 dos casos de entrada. Este parâmetro só é importante no caso de não ser igual em todos os PoPs, pois permite criar assimetrias no tráfego, simulando nós onde há servidores e nós onde há consumidores de informação. Para simplificar, colocou-se inicialmente este parâmetro igual em todos os PoPs.

Admitiu-se que o ritmo em todas as ligações era de 64 Kbps.

Admitiu-se que, para cada ligação, é usada apenas uma classe de tráfego, sendo a escolha por parte dos clientes aleatória, com todas as classes de tráfego igualmente prováveis. Para simplificar, admitiu-se que, das classes de envio assegurado, só era utilizada a classe AF1 para simular o tráfego normal, reservando-se as outras classes para utilizações específicas posteriores. Assim, utilizar-se-á a designação AF em vez de AF1, onde tal não originar confusão.

Colocou-se um condicionador em cada ligação que gera tráfego em EF ou AF, com um de dois perfis. O perfil para o tráfego EF descarta todos os pacotes que excedam o ritmo de pico negociado (64 Kbps). O perfil para o tráfego AF1 remarca os pacotes que excedam o ritmo médio negociado (64 Kbps) para o DSCP AF12, aumentando assim a sua probabilidade de descarte. Cada perfil funciona com um mecanismo de balde furado independente, tendo-se configurado o tamanho de cada balde para permitir rajadas de 32000 bits.

Na configuração da rede, utilizaram-se como configuração inicial os pesos indicados na tabela 5.1 (página 128) para as várias classes de tráfego, ficando portanto 50% da largura de banda garantida para EF, 40% da banda garantida para AF1, e apenas 10% garantido para BE.

As filas de espera de EF foram configuradas com uma dimensão de 35 pacotes.

As filas de espera de AF1 foram configuradas com uma dimensão de 120 pacotes, sendo os limiares do RIO os indicados na tabela 5.2. Estes limiares foram escolhidos de acordo com o exposto na secção 5.2.4.2.

As filas de espera de BE foram configuradas com uma dimensão de 180 pacotes, sendo os limiares do RED os indicados na tabela 5.3.

|  | Tráfego fora de perfil                            | Tráfego dentro de perfil |
|--|---|--------------------------|
| <b>Dimensão</b>                        | 120   |                          |
| $w_q$                                  | 0.01  |                          |
| <b>Comprimento Médio calculado com</b> | tráfego fora de perfil + tráfego dentro de perfil | tráfego dentro de perfil |
| $P_{máx}$                              | 1/10  | 1/50                     |
| <b>Limiar Máximo</b>                   | 30  | 60                       |
| <b>Limiar Mínimo</b>                   | 15  | 30                       |

Tabela 5.2: Configuração das filas de espera de AF.

|                      |      |
|----------------------|------|
| <b>Dimensão</b>      | 180  |
| $w_q$                | 0.01 |
| $P_{máx}$            | 1/20 |
| <b>Limiar Máximo</b> | 90   |
| <b>Limiar Mínimo</b> | 45   |

Tabela 5.3: Configuração das filas de espera de BE.

Uma vez que se pretende que a classe EF tenha menores atrasos do que a classe AF, e esta, por sua vez, tenha menores atrasos do que a classe BE, será também natural que a dimensão da fila EF seja menor do que o limiar máximo para o tráfego AF dentro de perfil, que deve, por sua vez, ser menor do que o limiar máximo para o tráfego BE.

Para o controlo de admissão de ligações utilizaram-se corretores de largura de banda com reservas dinâmicas. Para a configuração dos corretores de largura de banda, procedeu-se de acordo com o descrito na secção 5.2.3.1. Há dois pontos limitativos na rede: o núcleo da rede, onde o corte mínimo tem a capacidade de uma linha (10 Mbps), e cada PoP, onde há uma limitação correspondente à capacidade da linha de acesso à rede (2 Mbps). Assim, quando um cliente pretende estabelecer uma ligação em EF ou AF consulta o seu corretor de largura de banda local, que por sua vez tem de consultar o corretor responsável pelo núcleo da rede e o corretor responsável pelo PoP de destino da ligação. Se os três concordarem que há largura de banda disponível, a ligação pode ser estabelecida. Caso contrário, a ligação pode ser estabelecida numa classe abaixo, eventualmente BE, ou o utilizador pode desistir da ligação. Nas simulações realizadas, admitiu-se sempre que o utilizador queria que a ligação fosse estabelecida, aceitando uma classe abaixo da pedida, se necessário. Admitiu-se ainda, para simplificar, que todas as reservas eram bidireccionais com o mesmo ritmo em ambos os sentidos.

Como ficou patente tanto pela figura 4.9, como pela figura 4.10 da secção 4.3.3, é normal numa rede o tráfego ter grandes variações ao longo do tempo. Por esta razão, para cada política activa, avaliou-se o desempenho com diferentes valores do tráfego oferecido à rede. Como as linhas de acesso aos PoPs são de 2 Mbps e cada ligação de 64 Kbps, cada PoP pode ter 31 ligações simultâneas sem haver sobrecarga. Sendo igualmente prováveis as ligações geradas num PoP e as ligações de outros PoPs para ele, temos que o valor da carga média gerada em cada PoP que não causa sobrecarga é de 15 ligações. Mas será natural esperar que, se houver muitas ligações com o mesmo destino (ou origem), o limite seja atingido antes para alguns PoPs. Assim, fizeram-se simulações com valores de carga por PoP até cerca de 8 vezes esta carga limite para testar como as políticas activas se comportavam mesmo nas situações de sobrecarga.

Nos vários gráficos que serão apresentados, marcou-se, para cada ponto, o intervalo de confiança a 95% com uma barra horizontal acima e outra abaixo do ponto que representa a média de várias simulações. Os intervalos de confiança foram calculados com base no exposto no apêndice C, tendo sido realizadas 10 simulações



para cada ponto, utilizando sementes diferentes para os números aleatórios. Em alguns casos, os intervalos de confiança são tão pequenos que não se distinguem do próprio ponto correspondente à média.

#### 5.3.4 Discussão

Os parâmetros de simulação descritos na subsecção anterior não correspondem totalmente a uma situação real na Internet. De facto, na Internet actual, o tráfego depende muito da zona em causa, da hora, e mesmo da direcção que se observa.

Estudos sobre o tráfego na Internet [Floyd 01] demonstram que a utilização da distribuição exponencial é adequada para modelar o intervalo entre chegadas de novos utilizadores, correspondendo à decisão de uma pessoa começar a utilizar a rede para uma tarefa específica. Uma vez que cada chegada corresponde, normalmente, a um novo utilizador, é natural que o comportamento se aproxime do comportamento de um processo sem memória. Naturalmente que a taxa de chegadas deve variar de acordo com a hora do dia, tal como visto anteriormente na secção 4.3.3. No entanto, a distribuição exponencial já não é adequada para modelar a duração das ligações, ou o tráfego dentro das ligações.

De acordo com este estudo [Floyd 01], a duração de uma ligação, ou o tráfego que ela transporta, são adequadamente descritas pela família de distribuições log-normal. Isto é, a distribuição dos logaritmos das durações de ligações ou do tráfego transportado são bem aproximados pela distribuição gaussiana. Para caracterizar distribuições associadas com a actividade na rede, deve-se esperar distribuições com cauda pesada (*heavy-tailed*). É mais fácil de compreender o significado destas distribuições com um exemplo [Paxson 95]. Considerando uma variável aleatória  $X$  que representa o tempo de espera, a cauda é a probabilidade de esperar mais do que um certo tempo ( $P[X > x]$ ). A cauda é leve se, quanto mais se esperou, mais provável se torna estar despachado brevemente. A cauda é média em casos, como a distribuição exponencial (sem memória), em que o tempo de espera futuro é independente do tempo que já se esperou. Por outro lado, a cauda é pesada se, quanto mais se esperou, mais provável é ainda ter de se esperar mais. Um exemplo de uma distribuição com cauda pesada é a distribuição de Pareto com parâmetro de forma  $\alpha < 2$ . Verifica-se que distribuições com cauda pesada são um modelo adequado para fenómenos tais como

[Floyd 01]: o tempo de processamento gasto por processos Unix; o tamanho de ficheiros Unix, tramas de vídeo comprimidas e objectos WWW; e rajadas de actividade Ethernet e FTP.

Este estudo [Floyd 01] ainda refere que as correlações de longo prazo nas chegadas de pacotes observadas no tráfego agregado na Internet são bem modeladas por processos fractais, ou auto-similares (*self-similar*). Isto significa que o tráfego parece semelhante em diferentes escalas de tempo. Uma forma possível de gerar tráfego com propriedades fractais é [Paxson 95] multiplexar tráfego de fontes OnOff com ritmo fixo nos períodos On e períodos tirados de uma distribuição com cauda pesada.

Desta forma, as simulações realizadas, por utilizarem principalmente a distribuição exponencial, correspondem a uma aproximação simplificada da realidade da rede Internet. No entanto, procurou-se utilizar diversos tipos de fontes de tráfego de forma a melhor aproximar o funcionamento de uma rede real.

Outro aspecto tem a ver com a distribuição de protocolos utilizada, que também não corresponde exactamente à situação real. A tabela 5.4 [Edell 99] indica a distribuição por protocolo do tráfego dos utilizadores num ISP e no sistema INDEX. A primeira coluna indica a classificação do protocolo de acordo com a utilização no ISP. A terceira coluna é a percentagem do volume de tráfego por protocolo para o ISP, e a quarta coluna para o sistema INDEX. O sistema INDEX é um ISP experimental usado para investigação na universidade da Califórnia em Berkeley. Da comparação dos dois sistemas, verifica-se que as quatro aplicações mais utilizadas no ISP (WWW, News, email e FTP) correspondem a 96% do tráfego, enquanto os utilizadores do INDEX utilizam uma gama mais larga de protocolos. Isso pode ser explicado pelo facto de os utilizadores do INDEX serem mais experientes.

Também os ritmos usados nas simulações não correspondem aos ritmos da Internet actual. Na realidade, o núcleo da rede de alguns operadores chega a ter linhas OC-192c a 10 Gbps, prevendo-se 40 Gbps para 2002 [Liu 01], sendo natural que os ritmos continuem a aumentar com a vulgarização da tecnologia WDM (*Wavelength-Division Multiplexing*) [Ryan 98]. Além disso, com acessos por *Digital Subscriber Line* (DSL), muitos utilizadores individuais já têm ritmos superiores a 64 Kbps, sendo os ritmos muito superiores para empresas. Também nas redes locais os ritmos de 100 Mbps são normais, estando já disponível a Ethernet a 1 Gbps e a norma da Ethernet a 10 Gbps em preparação [Baritault 00].

| Lugar          | Protocolo  | ISP (%)        | INDEX (%) |
|----------------|------------|----------------|-----------|
| 1              | HTTP       | 76             | 43        |
| 2              | NNTP       | 12             | 2         |
| 3              | POP3       | 3              | 3         |
| 4              | FTP        | 3              | 11        |
| 5              | SMTP       | 2              | 0.5       |
| 6              | HTTPS      | 2              | 2         |
| 18             | Telnet/SSH | 0              | 4         |
| Não Disponível | X-Windows  | Não Disponível | 2         |

Tabela 5.4: Distribuição da utilização de protocolos num ISP e no sistema INDEX.

No entanto, o mais importante é manter uma certa relação entre os ritmos nas ligações e nas linhas do núcleo da rede, e não tanto os valores absolutos destes ritmos. A utilização de ritmos superiores tem como principal efeito exigir uma maior capacidade de processamento e mais memória para guardar informação. Todo o processamento será semelhante, havendo apenas um maior número de eventos a processar por unidade de tempo. Assim, será possível fazer simulações utilizando ritmos relativamente baixos, sem perder generalidade, desde que os tempos de resposta aos eventos e as durações das simulações sejam ajustados de acordo com os ritmos de ocorrência de eventos na rede.

Relativamente à duração das simulações, verificou-se que, configurando as simulações para começar com uma carga inicial correspondente ao valor pretendido para o tráfego oferecido, o período transitório no início das simulações era muito pouco significativo, pelo que se obtinha praticamente os mesmos resultados simulando apenas um minuto de tempo real, ou fazendo uma simulação mais longa. Na realidade, sendo o efeito do transitório inicial desprezável, o que influi nos resultados é principalmente o valor médio do tráfego oferecido à rede, que se fez constante em cada simulação. Assim, optou-se por fazer simulações de apenas um minuto de tempo real, variando-se a carga de simulação para simulação, de forma a analisar diferentes situações de carga. Isto significa que os tempos de reacção de algumas políticas activas têm de ser mais curtos do que na realidade seriam, para que se possa observar o seu efeito. Para os casos em que se utilizaram políticas activas de planeamento, fizeram-se simulações mais longas, de três minutos de tempo real, de forma a que as políticas activas tivessem tempo de reagir eficazmente. Também neste caso se verificou que os resultados obtidos eram

semelhantes aos de uma simulação mais longa, não se justificando alongar mais as simulações.

Foram feitas outras simplificações na simulação. Por exemplo: o tempo está sempre sincronizado em todos os nós da rede, visto ser o tempo do simulador. Num sistema real, esta propriedade pode ser obtida, tal como sugerido em [RFC 2330], pela utilização do protocolo NTP (*Network Time Protocol*), o que pode ser importante, por exemplo, para a medição de atrasos unidireccionais. Outra simplificação advém do facto de não haver nenhum protocolo de sinalização com os corretores de largura de banda. Isto significa que o tráfego correspondente é desprezado em face do tráfego de dados. Também não é simulado o tráfego de comunicação entre as políticas activas. O cenário que mais se aplica a estas decisões é as políticas activas estarem colocadas junto aos equipamentos que monitorizam, de forma a não utilizarem largura de banda da rede, e as notificações que enviam entre elas, por serem raras, poderem ser desprezadas face ao tráfego de dados. Em termos de largura de banda, estas simplificações podem ser anuladas sobredimensionando a capacidade da rede ligeiramente, para levar em conta estes tipos de tráfego. Para além disso, em termos de tempo, numa rede real vai haver também atrasos na transmissão do tráfego de gestão que podem causar problemas. Uma forma de minimizar os seus efeitos seria enviar as notificações de problemas, a que se tivesse de reagir rapidamente, como tráfego de prioridade máxima, desde que tal seja suportado pela rede. No entanto, esses atrasos na transmissão devem ser sempre muito menores do que o tempo que o sistema demora a evoluir, de forma a que se possa reagir a tempo e no sentido correcto.

Recorde-se, para terminar, que o foco deste trabalho está em testar as potencialidades de uma infra-estrutura de gestão baseada em políticas activas, a metodologia necessária à sua implementação, e assim validar o modelo proposto nesta tese. Os impactos das simplificações realizadas devem ser vistos nesta perspectiva e não numa perspectiva de fidelidade estrita do cenário à realidade da Internet.

## **5.4 APLICAÇÃO DO MODELO - POLÍTICAS ACTIVAS**

A figura 5.15 ilustra as políticas activas identificadas para cada nível, e como elas se relacionam entre si. Esta arquitectura e as principais políticas activas foram

descritas em [Pereira 01a]. Nesta secção descreve-se detalhadamente o funcionamento de cada uma individualmente, deixando-se para a secção seguinte a situação de todas as políticas activas a funcionar simultaneamente. Estas duas secções correspondem à segunda e terceira fase da metodologia para a construção de um sistema de gestão por políticas activas, conforme proposto anteriormente na secção 3.5.

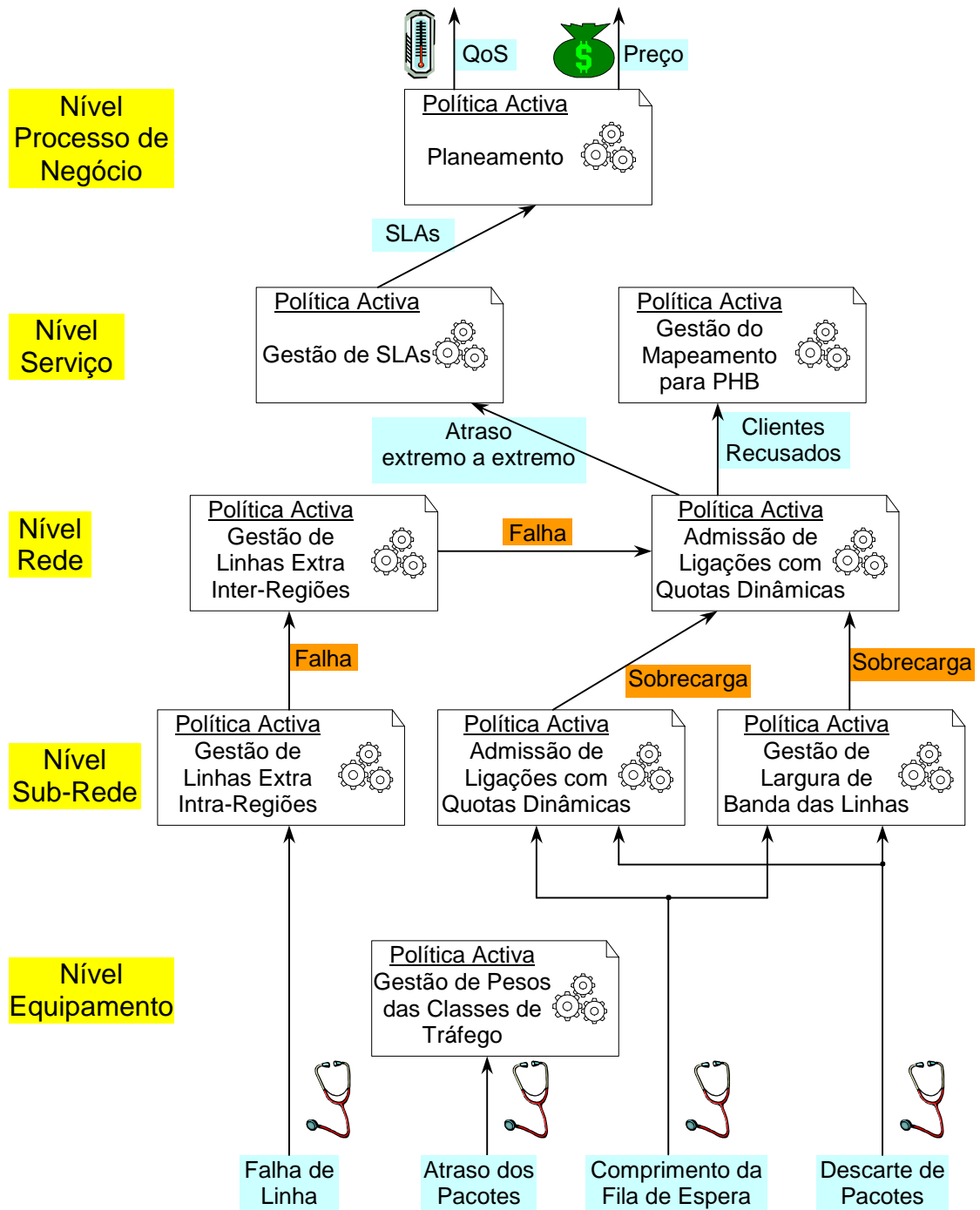


Figura 5.15: Políticas activas utilizadas e as suas relações.

### 5.4.1 Política Activa de Gestão dos Pesos das Classes de Tráfego

As políticas activas de gestão dos pesos das classes de tráfego têm como objectivo ajustar os pesos das diversas classes de tráfego de forma a que a qualidade de serviço (definida, por exemplo, pelo atraso, *jitter* e débito) seja a melhor para a classe EF, seguida pelas classes AF começando pela de maior prioridade, e seja pior para a classe BE. Além disso, estas políticas ainda ajustam os limiares máximo e mínimo de descarte do tráfego AF fora de perfil, como forma de controlar a quantidade de tráfego AF fora de perfil que é transportada pela rede, aumentando o seu descarte no caso de sobrecarga da linha.

Estas políticas funcionam localmente a cada linha, de forma independente, monitorizando continuamente os atrasos experimentados pelos pacotes nas classes EF, AF e BE. Estes atrasos são comparados entre si e, quando possível, com o SLA. Se bem que não seja possível a este nível saber qual a contribuição de cada linha para o atraso extremo a extremo especificado no SLA, e assim assegurar o seu cumprimento, é possível saber se uma dada linha só por si está a exceder o atraso especificado no SLA, caso em que se deve reagir descartando o tráfego AF fora de perfil nessa linha. Além disso, estas políticas podem receber indicação, de uma política num nível acima, do limiar do atraso a partir do qual devem descartar o tráfego AF fora de perfil.

Esta política calcula e disponibiliza quatro parâmetros de qualidade de serviço:

O primeiro parâmetro,  $QoS(atrasoEF)$ , dá uma indicação do atraso que os pacotes EF estão a sofrer ao passar na fila de espera e na linha onde a política actua. Um valor maior para este parâmetro de QoS representa uma situação pior.

O segundo parâmetro,  $QoS(atrasoAF)$ , é o equivalente do primeiro para os pacotes AF.

O terceiro parâmetro,  $QoS(atrasoBE)$ , é o equivalente do primeiro para os pacotes BE.

O quarto parâmetro,  $QoS(atraso)$ , dado por:

$$QoS(atraso) = \frac{\frac{atrasoEF}{atrasoAF}}{\frac{SLA(atrasoEF)}{SLA(atrasoAF)}} \quad (5.3)$$

representa a fracção entre o atraso verificado na classe EF e o atraso verificado na classe AF, relativamente ao que devia ser obtido, de acordo com o SLA. Pretende-se que este

parâmetro de QoS seja 1.0, ou inferior. Isto é, que o tráfego EF não seja prejudicado relativamente ao tráfego AF.

O peso da classe EF é ajustado entre dois valores limites de acordo com a regra *threshold\_increment* aplicada ao valor de *QoS(atraso)* a intervalos de 25 milissegundos. Limitou-se a variação do peso da classe EF entre 5 e 12, correspondendo respectivamente a 50% e 71% da largura de banda da linha que pode assim ficar reservada para EF. O funcionamento desta regra está ilustrado na figura 5.16, com indicação dos valores utilizados para os diversos parâmetros. Há três zonas de funcionamento. Na zona em que a QoS é alta, correspondente a valores superiores a 1.0, isto é: em que o atraso da classe EF é maior do que o pretendido, aumenta-se o peso da classe EF de duas unidades, até se atingir o valor máximo possível. Na zona em que a QoS é normal, correspondente a valores entre 0.5 e 1.0, mantém-se o peso da classe EF. Na zona em que a QoS é baixa, correspondente a valores inferiores a 0.5, baixa-se o peso da classe EF de duas unidades, até atingir o valor mínimo possível.

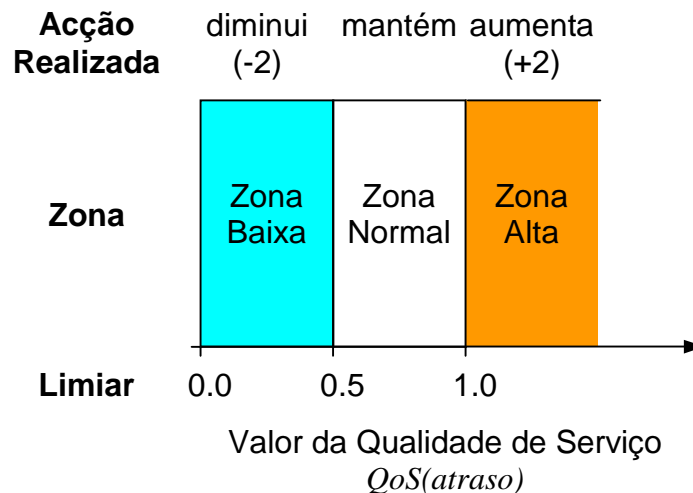


Figura 5.16: Funcionamento da regra de ajuste do peso da classe EF.

Escolheu-se um intervalo curto entre ajustes consecutivos para permitir uma melhor adaptação às características do tráfego. É possível ter um intervalo desta duração pois esta política é local a cada linha, não dependendo de nenhum tráfego de gestão.

O peso da classe AF é ajustado entre dois valores limites de acordo com a mesma regra *threshold\_increment*, mas parametrizada de forma diferente. Agora pretende-se que o atraso em AF seja inferior ao de BE. Assim, a regra é aplicada ao

valor de  $QoS(atrasoAF)/QoS(atrasoBE)$  também a intervalos de 25 milissegundos. Limitou-se a variação do peso da classe AF entre 4 e 6, o que representa de 4 a 6 vezes a largura de banda da linha que está reservada para BE. O funcionamento desta regra está ilustrado na figura 5.17 com indicação dos valores utilizados para os diversos parâmetros. Na zona em que a  $QoS(atrasoAF)$  é superior a  $QoS(atrasoBE)$ , correspondente a valores da QoS superiores a 1.0, aumenta-se o peso da classe AF de uma unidade, até se atingir o valor máximo possível. Na zona em que a QoS é normal, correspondente a valores entre 0.8 e 1.0, mantém-se o peso da classe AF. Na zona em que a QoS é baixa, correspondente a valores inferiores a 0.8, baixa-se o peso da classe AF de uma unidade, até atingir o valor mínimo possível.

Uma vez que a classe EF deve ter melhor QoS do que a classe AF, na regra anterior altera-se o peso da classe EF de duas unidades de cada vez, enquanto nesta regra altera-se o peso da classe AF de apenas uma unidade de cada vez. Assegura-se, desta forma, a maior importância da classe EF.

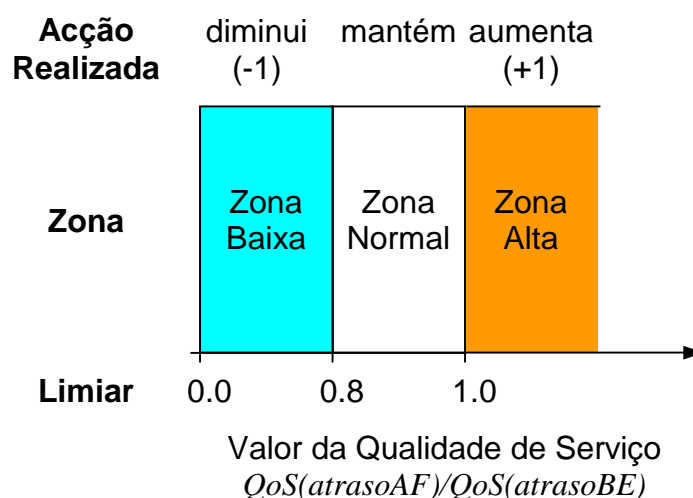


Figura 5.17: Funcionamento da regra de ajuste do peso da classe AF.

O descarte de pacotes AF fora de perfil é controlado de acordo com o atraso que esses mesmos pacotes sofrem, conforme um dos casos descritos na tabela 5.5. O parâmetro *AFOutPercent* determina a fracção do SLA que, quando ultrapassada, causa o descarte dos pacotes AF fora de perfil, sendo o seu valor normal 1.0. O primeiro caso da tabela 5.5 é o caso normal, em que o atraso sofrido pelos pacotes AF é inferior ao limite imposto pelo SLA, pelo que se utilizam os limiares normais no RIO. No segundo caso,



o atraso dos pacotes AF na linha controlada por esta política excede o atraso máximo imposto pelo SLA para o atraso extremo a extremo, pelo que todo o tráfego AF fora de perfil é descartado. Naturalmente que este segundo caso só acontece em situações extremas, pelo que outras políticas activas devem agir antes. O parâmetro *AFOutPercent* é modificado pela política activa de gestão de SLAs, que funciona no nível serviço, tendo uma visão extremo a extremo dos parâmetros do SLA, e permitindo assim definir qual o limiar a partir do qual o tráfego fora de perfil deve ser descartado.

| <b>Caso</b>                                       | <b>Limiar Mínimo</b> | <b>Limiar Máximo</b> |
|---|----------------------|----------------------|
| $atrasoAF < SLA(atrasoAF) \times AFOutPercent$    | 15                   | 30                   |
| $atrasoAF \geq SLA(atrasoAF) \times AFOutPercent$ | 0                    | 0                    |

Tabela 5.5: Limiares de descarte de pacotes AF fora de perfil.

Para testar o funcionamento desta política activa, utilizou-se para o  $SLA(atrasoEF)$  o valor de 60 ms, para  $SLA(atrasoAF)$  o valor de 120 ms, e para *AFOutPercent* sempre o valor normal de 1.0.

A figura 5.18 ilustra o atraso médio (em segundos) sofrido pelos pacotes extremo a extremo nas diferentes classes de tráfego, em função da carga de ligações oferecida à rede, com e sem a política activa de gestão dos pesos das classes de tráfego. A figura 5.19 ilustra a fracção de pacotes descartados, que é nula para o tráfego EF (não representado) e para o tráfego AF dentro de perfil, sendo apenas não nula para o tráfego AF fora de perfil e BE. Verifica-se que o efeito da regra que descarta tráfego AF fora de perfil é muito mais importante que o efeito das regras que gerem os pesos das classes EF e AF. Na realidade, nas situações em que se pretende que a rede funcione, a carga nas classes EF e AF é limitada pelo corretor de largura de banda, sendo relativamente reduzida, pelo que as filas de espera estão a maior parte do tempo quase vazias. Além disso, os atrasos na transmissão e propagação dos pacotes dependem apenas da linha, sendo, por isso, os mesmos para todas as classes de tráfego. Assim sendo, os valores dos pesos das classes são pouco importantes nestas situações, pelo que o efeito desta política activa também é pouco importante, já que os pacotes são servidos aproximadamente ao ritmo a que chegam à fila de espera. Quando ocorre uma sobrecarga na classe AF (a situação crítica mais real), o descarte de pacotes fora de perfil é muito mais eficaz para aliviar a sobrecarga. Só se ocorrer simultaneamente uma sobrecarga em EF e AF é que o

mecanismo de ajuste dos pesos tem algum efeito, continuando o mecanismo de descarte de pacotes AF fora de perfil a ter maior efeito. Desta forma, verifica-se que o atraso na classe EF praticamente não é modificado, enquanto o atraso na classe AF melhora significativamente, visto que há linhas onde o atraso na classe AF ocasionalmente excede os 120 ms, causando o descarte do tráfego fora de perfil. Como consequência, o atraso da classe BE também melhora um pouco, pois sobram mais recursos da classe AF.

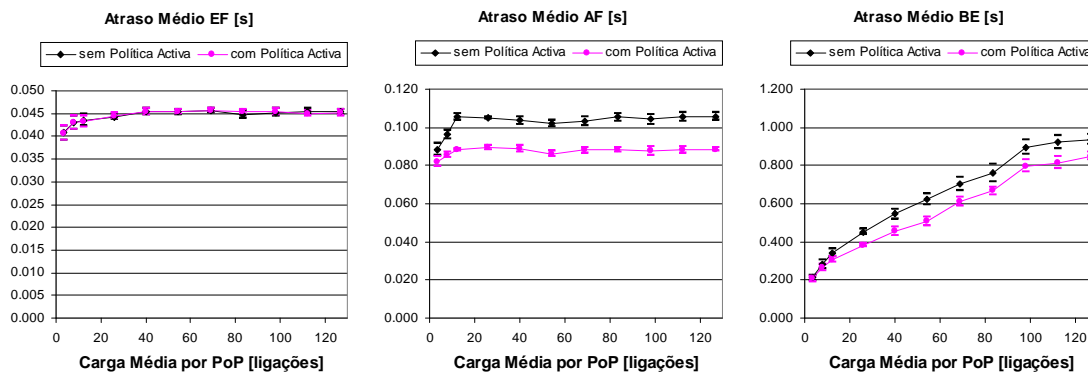


Figura 5.18: Comparação do atraso médio com e sem a política activa de gestão dos pesos das classes de tráfego.

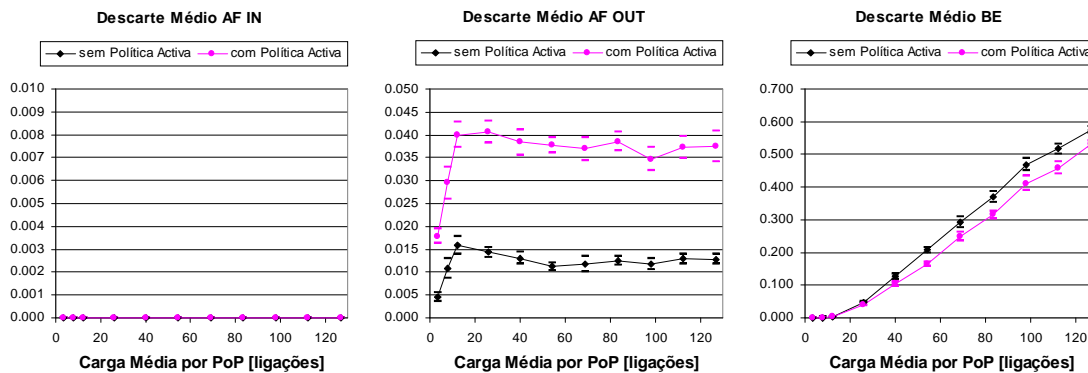


Figura 5.19: Comparação do descarte de pacotes com e sem a política activa de gestão dos pesos das classes de tráfego.

Se bem que o efeito desta política sobre a classe EF não seja significativo, esta política é útil pois permite o controlo do descarte do tráfego AF fora de perfil por parte de políticas de mais alto nível, através do parâmetro *AFOutPercent* que define a partir de que fracção do SLA do atraso de AF é que se descarta o tráfego fora de perfil.

Permite-se assim descartar o tráfego AF fora de perfil apenas nas linhas que mais contribuem em cada momento para exceder o SLA.

#### 5.4.2 Política Activa de Gestão de Largura de Banda das Linhas

As políticas activas de gestão de largura de banda das linhas têm como objectivo adaptar a largura de banda das linhas do núcleo da rede às necessidades de largura de banda de cada momento. Para isso, admite-se, tal como descrito na subsecção 5.2.3.2, que a largura de banda das linhas pode ser ajustada entre certos limites. Admitiu-se ainda que este ajuste da largura de banda podia ser feito de forma independente nas duas direcções.

Estas políticas são locais a cada linha do núcleo da rede, monitorizando continuamente para as classes EF e AF os descartes de pacotes e o comprimento da fila de espera, de forma a tomar decisões sobre a largura de banda pretendida para a linha. Caso o algoritmo decida que a largura de banda deve ser aumentada, e isso não seja possível, é gerada uma notificação de sobrecarga.

Esta política calcula três parâmetros de qualidade de serviço:

$$\begin{aligned}
 QoS(descartes) = & \\
 & \text{número de descartes de pacotes em EF} + \\
 & \text{número de descartes de pacotes dentro de perfil em AF} \quad (5.4)
 \end{aligned}$$

$$\begin{aligned}
 QoS(fila) = & \\
 & \text{comprimento da fila EF} \times 3 / \text{comprimento máximo da fila EF} + \\
 & \text{comprimento da fila AF} \times 4 / \text{comprimento máximo da fila AF} \quad (5.5)
 \end{aligned}$$

$$\begin{aligned}
 QoS(largura\ de\ banda) = & \\
 & \text{largura de banda actual} / \text{largura de banda máxima admitida} \quad (5.6)
 \end{aligned}$$

O primeiro parâmetro,  $QoS(descartes)$ , dá uma indicação do número de pacotes descartados. Só se contam os descartes de pacotes EF e de pacotes AF dentro de perfil, pois são os pacotes que não devem ser descartados pela rede. Um valor maior para este parâmetro de QoS representa uma situação pior. Pretende-se que este parâmetro seja zero.

O segundo parâmetro,  $QoS(fila)$ , dá uma indicação da ocupação das filas de espera, e do conseqüente atraso sofrido pelos pacotes. Neste caso, o comprimento da fila AF inclui tanto pacotes dentro, como fora de perfil. Utilizou-se o comprimento da fila, e não o comprimento médio, para garantir respostas mais rápidas a situações de carga, e mais facilmente assegurar os requisitos de QoS. No entanto, torna-se necessário utilizar regras de actuação que prevejam o caso de variações rápidas do comprimento da fila, devidas a rajadas de pacotes. Para as filas de espera de AF, configuraram-se os limiares de descarte tal como sugerido na secção 5.2.4.2, pelo que a parcela de  $QoS(fila)$  correspondente a AF vale 0.5 quando se atinge o limiar mínimo para pacotes fora de perfil, 1.0 quando se atinge o limiar máximo para pacotes fora de perfil, que coincide com o limiar mínimo para pacotes dentro de perfil, 2.0 quando se atinge o limiar máximo para pacotes dentro de perfil, e 4.0 quando a fila de espera transborda. Como facilmente se nota, um valor maior para este parâmetro de QoS corresponde a maiores atrasos verificados pelos pacotes, o que representa uma situação pior. A constante da fórmula foi escolhida para dar o valor 1.0 no limite da situação normal, em que ainda não há descarte de pacotes AF dentro de perfil. A constante 3 na parcela correspondente a EF tem como objectivo dar um peso à classe EF três vezes superior ao da classe AF, visto que só há descarte de pacotes em EF quando se atingir o comprimento máximo da fila de espera.

O terceiro parâmetro,  $QoS(largura\ de\ banda)$ , é a fracção da largura de banda máxima possível que actualmente está requisitada para uso, dando uma indicação da distância do ponto de funcionamento actual ao limite máximo possível. Um valor maior para este parâmetro de QoS representa custos acrescidos para o operador, mas maior qualidade de serviço (atrasos, débito) para os utilizadores. Maiores custos para o operador pode significar a utilização de mais largura de banda para este serviço, deixando de estar disponível para outros serviços, ou o aluguer de largura de banda a outro operador.

A largura de banda é ajustada entre dois valores limites de acordo com a regra `adjust_increment` aplicada ao valor de  $QoS(descartes)+QoS(fila)$  a intervalos de 25 ms. O funcionamento desta regra está ilustrado na figura 5.20 com indicação dos valores utilizados para os diversos parâmetros. Há cinco zonas de funcionamento. Na zona segura, correspondente a valores da QoS inferiores a 0.4, a largura de banda é diminuída lentamente de 0.5 Mbps de cada vez, correspondendo a 5% do valor da

largura de banda do caso fixo, até atingir o valor mínimo possível. Na zona normal, correspondente a valores entre 0.4 e 1.0, não é modificada a largura de banda da linha. Na zona de perigo, correspondente a valores entre 1.0 e 1.5, a largura de banda é aumentada de 1 Mbps de cada vez, correspondendo a 10% do valor da largura de banda do caso fixo, até atingir o valor máximo permitido. Na zona de perigo grave, correspondente a valores entre 1.5 e 2.5, a largura de banda é duplicada. Finalmente, na zona de pânico, correspondente a valores acima de 2.5, a largura de banda é colocada no valor máximo possível.

Esta regra aumenta a largura de banda rapidamente sempre que há uma situação de degradação da QoS. Por outro lado, diminui lentamente a largura de banda quando a situação volta ao normal, como forma de evitar oscilações e não pôr em perigo a manutenção da QoS pretendida, tal como explicado anteriormente na secção 3.5.3.

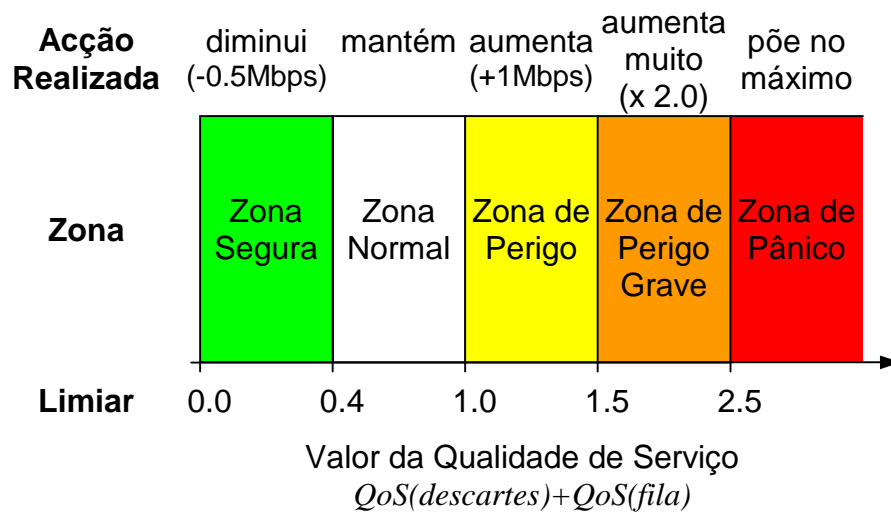


Figura 5.20: Funcionamento da regra de ajuste da largura de banda.

Uma segunda regra aumenta a largura de banda de 1 Mbps caso o  $SLA(atrasoEF)$ , ou o  $SLA(atrasoAF)$ , estejam a ser violados. Esta regra tem o mesmo efeito para todas as linhas, visto que não depende de nenhum parâmetro da linha sob gestão, mas só actua por indicação de uma política num nível superior que dá a indicação se os SLAs estão, ou não, a ser cumpridos. Nos resultados desta secção, esta regra nunca é activada, visto que se está a analisar cada política activa em separado, deixando-se para a secção seguinte (5.5) o caso das diferentes políticas activas a funcionar simultaneamente.

Caso tenha havido uma tentativa de aumentar a largura de banda que tenha falhado, e a largura de banda utilizada seja pelo menos 90% da máxima possível, correspondendo a  $QoS(largura\ de\ banda) \geq 0.90$ , então é gerada uma notificação de sobrecarga.

Convém notar que o valor escolhido para limite máximo da largura de banda nas linhas do núcleo é o novo valor do corte mínimo do núcleo que é utilizado no corretor de largura de banda do núcleo. Isto significa que pode ser aceite um maior número de ligações, como será visto mais à frente nesta subsecção.

No entanto, isto pode levantar o problema de a largura de banda das linhas demorar algum tempo a ser modificada. Caso se admita que a fracção da largura de banda relativa a EF pode ser modificada, deixa de haver garantia de não haver descarte de pacotes em EF, apesar de as reservas terem sido feitas no corretor de largura de banda. Este facto pode ser resolvido de uma de duas maneiras: ou a fracção de largura de banda de EF é calculada directamente a partir da largura de banda que está reservada no corretor de largura de banda em cada momento, ou então é necessário aumentar a dimensão dos *buffers* de EF para dar tempo a que a largura de banda das linhas seja modificada quando for necessário, sem descarte de pacotes. A primeira solução exige mais tráfego de gestão entre o corretor de largura de banda e as políticas activas que gerem a largura de banda das linhas. Assim, optou-se pela segunda solução, que torna estas políticas activas independentes, apesar de resultar num aumento adicional no *jitter*. Caso se tivesse optado por manter sempre fixa a fracção da largura de banda relativa a EF, as economias na largura de banda utilizada seriam muito menores.

De seguida, vai-se calcular a dimensão requerida para o *buffer* de EF. Sendo  $\Delta t$  o intervalo de tempo necessário para modificar a largura de banda de uma linha, que se admitiu anteriormente ser no máximo de 25 ms, a dimensão da rajada  $L$  em pacotes que é necessário suportar deve ser de:

$$L > \frac{\Delta t}{\frac{\text{Dimensão Pacote}}{R_{\text{máx}} - R_{\text{inicial}}}} \quad (5.7)$$

que corresponde ao número de pacotes extra que podem chegar, se o ritmo fosse o máximo, enquanto não se obtém a modificação da largura de banda.

Decidiu-se permitir que esta política activa regulasse a largura de banda de cada linha do núcleo da rede entre 2 Mbps e 30 Mbps.

Utilizando como dimensão do pacote 1000 bytes, como fracção da largura de banda reservada para EF 50%, e considerando como ocupação inicial do *buffer* os casos limites superiores das zonas normal, de perigo e de perigo grave da figura 5.20, calculou-se na tabela 5.6 as dimensões necessárias correspondentes para o *buffer* de EF. Por este cálculo, conclui-se que o *buffer* de EF deve ter 70 pacotes para evitar descartes em qualquer das situações, ou seja, o dobro da dimensão que seria utilizada sem esta política activa. O mesmo cálculo pode ser feito em relação ao *buffer* de AF, sendo neste caso a fracção de largura de banda reservada de 40%. Os resultados estão indicados na tabela 5.7, verificando-se que, neste caso, a tolerância já existente é suficiente. Ainda assim, optou-se por aumentar o *buffer* de AF pelo valor resultante do cálculo, ou seja, de 35 pacotes.

| Ocupação Inicial | <i>QoS(fila)</i>         | <i>Rinicial</i> | <i>Rmáx</i> | <i>L</i> | Dimensão Buffer |
|------------------|--------------------------|-----------------|-------------|----------|-----------------|
| 11               | $3 \times 11 / 35 < 1$   | 1 Mbps          | 15 Mbps     | 44       | 55              |
| 17               | $3 \times 17 / 35 < 1.5$ | 1.5 Mbps        | 15 Mbps     | 42       | 59              |
| 29               | $3 \times 29 / 35 < 2.5$ | 2 Mbps          | 15 Mbps     | 41       | 70              |

Tabela 5.6: Dimensionamento do *buffer* de EF.

| Ocupação Inicial | <i>QoS(fila)</i>          | <i>Rinicial</i> | <i>Rmáx</i> | <i>L</i> | Dimensão Buffer |
|------------------|---------------------------|-----------------|-------------|----------|-----------------|
| 29               | $4 \times 29 / 120 < 1$   | 0.8 Mbps        | 12 Mbps     | 35       | 64              |
| 44               | $4 \times 44 / 120 < 1.5$ | 1.2 Mbps        | 12 Mbps     | 34       | 78              |
| 74               | $4 \times 74 / 120 < 2.5$ | 1.6 Mbps        | 12 Mbps     | 33       | 107             |

Tabela 5.7: Dimensionamento do *buffer* de AF.

Começou-se por fazer uma simulação com um tráfego oferecido de 30 clientes por PoP, o que corresponde aproximadamente à capacidade da linha do PoP para o núcleo da rede com tráfego unidireccional (ver figura 5.13). A figura 5.21 ilustra a evolução do parâmetro *QoS(fila)*. Verificou-se que nunca houve descartes de pacotes nas classes EF e AF, pelo que se teve sempre  $QoS(descartes)=0$ , e o valor utilizado pela regra *adjust\_increment* foi sempre igual a *QoS(fila)*.

A figura 5.22 ilustra a evolução do parâmetro *QoS(largura de banda)*. Observa-se que há aumentos rápidos, correspondendo a situações em que a QoS está ameaçada, seguidos de diminuições lentas, até voltar ao normal. Verifica-se, ainda, que a largura de banda máxima nunca é atingida para este nível de carga da rede. Aliás, a largura de

banda média nas linhas do núcleo foi de apenas 2.837 Mbps, o que representa uma significativa economia de custos para o operador da rede.

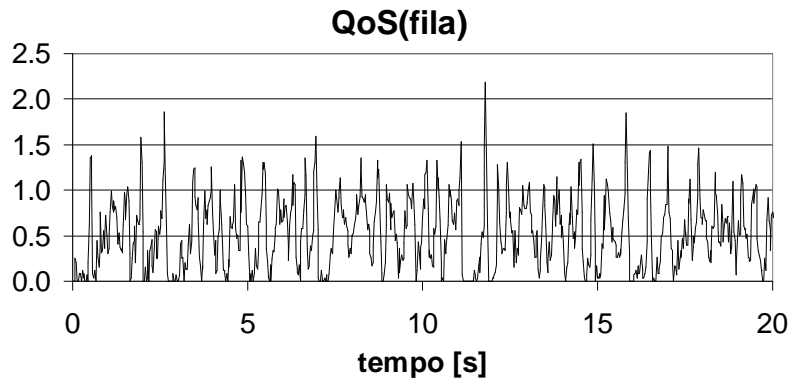


Figura 5.21: Evolução do parâmetro *QoS(fila)*.

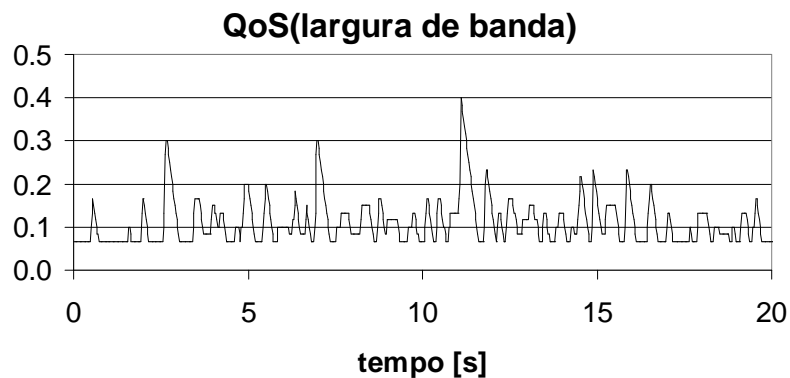


Figura 5.22: Evolução do parâmetro *QoS(largura de banda)*.

É interessante fazer variar a carga na rede para observar o efeito do funcionamento desta política activa. Na figura 5.23 ilustra-se, à esquerda, a largura de banda média (em bit/segundo) nas linhas do núcleo da rede com e sem a política activa de gestão da largura de banda, para diferentes valores do número médio de ligações entradas em cada PoP. Verificou-se que em nenhum dos casos houve descarte de pacotes AF dentro de perfil, nem de pacotes EF. Verificou-se ainda não ter havido nenhuma notificação de sobrecarga para as simulações efectuadas.

Posteriormente, verificou-se que nunca havia qualquer descarte de pacotes na classe EF, quando esta política activa era utilizada em conjunto com outras, ou não. Tal facto mostra que esta política activa é eficaz no cumprimento deste requisito.



Da observação da figura 5.23, verifica-se que a largura de banda necessária aumenta ligeiramente à medida que se aumenta a carga oferecida à rede, ficando sempre muito abaixo do valor fixo de 10 Mbps da situação sem política activa. Isto significa que nas situações normais, parte da largura de banda pode ser disponibilizada para outros serviços que possam utilizar a mesma infra-estrutura de linhas. Em qualquer dos casos, esta redução da largura de banda é feita mantendo a qualidade de serviço oferecida aos utilizadores dentro de limites aceitáveis.

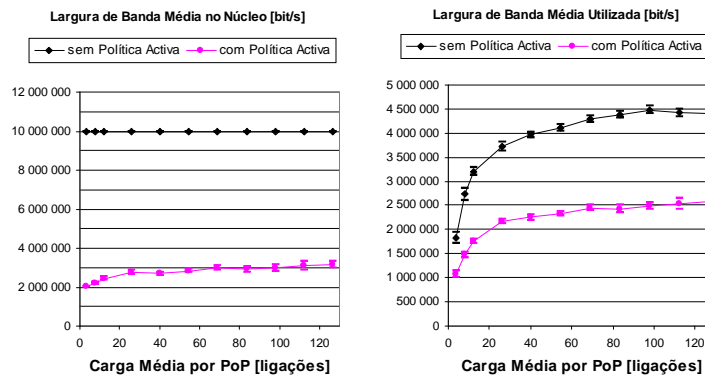


Figura 5.23: Comparação da largura de banda média e da largura de banda média utilizada com e sem a política activa de gestão de largura de banda.

O gráfico que está à direita na figura 5.23 ilustra a largura de banda (em bit/segundo) que foi, de facto, utilizada. Verifica-se que, no caso com a política activa, a largura de banda utilizada está mais próxima da largura de banda reservada, embora seja inferior à largura de banda utilizada no caso sem a política activa. Tal significa um melhor aproveitamento da largura de banda reservada, com a desvantagem de se deixar passar menos tráfego BE e AF fora de perfil, como se verá quando se analisar o débito das ligações.

Note-se, ainda, que uma duplicação da carga oferecida à rede não implica uma duplicação da largura de banda utilizada, pois, como se verá de seguida, há uma limitação no tráfego das classes EF e AF imposta pelo corretor de largura de banda.

A figura 5.24 ilustra o número médio de ligações nas diferentes classes de tráfego com e sem a utilização desta política activa. Verifica-se que para valores baixos da carga, o número de ligações na situação com política activa coincide com os valores obtidos sem a política activa. No entanto, para valores superiores da carga, o número de

ligações nas classes EF e AF é ligeiramente superior, mas não ultrapassa um dado valor máximo. Tal deve-se às limitações impostas pelo corretor de largura de banda, que impõe um limite ao número de ligações simultâneas nas classes EF e AF, conforme cálculo apresentado na tabela 5.8, onde  $\lfloor x \rfloor$  representa o maior inteiro não superior a  $x$ .

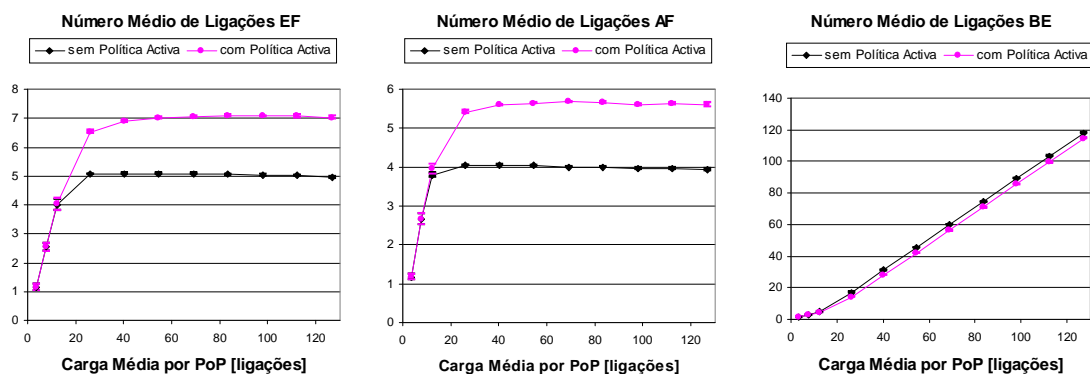


Figura 5.24: Comparação do número de ligações com e sem a política activa de gestão de largura de banda.

| Situação                   |           | sem Política Activa   | com Política Activa   |
|----------------------------|-----------|---|---|
| <b>Ritmo no Núcleo</b>     |           | 10 000 000 bps  | máximo 30 000 000 bps   |
| <b>Limitação no PoP</b>    | <b>EF</b> | $\left\lfloor \frac{2\,000\,000}{64\,000} \cdot 0.50 \right\rfloor = 15$<br>médio por PoP: 7.5  | $\left\lfloor \frac{2\,000\,000}{64\,000} \cdot 0.50 \right\rfloor = 15$<br>médio por PoP: 7.5    |
|                            | <b>AF</b> | $\left\lfloor \frac{2\,000\,000}{64\,000} \cdot 0.40 \right\rfloor = 12$<br>médio por PoP: 6.0  | $\left\lfloor \frac{2\,000\,000}{64\,000} \cdot 0.40 \right\rfloor = 12$<br>médio por PoP: 6.0    |
| <b>Limitação no Núcleo</b> | <b>EF</b> | $\left\lfloor \frac{10\,000\,000}{64\,000} \cdot 0.50 \right\rfloor = 78$<br>médio por PoP: 5.2 | $\left\lfloor \frac{30\,000\,000}{64\,000} \cdot 0.50 \right\rfloor = 234$<br>médio por PoP: 15.6 |
|                            | <b>AF</b> | $\left\lfloor \frac{10\,000\,000}{64\,000} \cdot 0.40 \right\rfloor = 62$<br>médio por PoP: 4.1 | $\left\lfloor \frac{30\,000\,000}{64\,000} \cdot 0.40 \right\rfloor = 187$<br>médio por PoP: 12.4 |
| <b>Menor</b>               | <b>EF</b> | médio por PoP: 5.2  | médio por PoP: 7.5  |
|                            | <b>AF</b> | médio por PoP: 4.1  | médio por PoP: 6.0  |

Tabela 5.8: Limites utilizados no corretor de largura de banda.

As diferenças entre as duas situações são devidas ao facto de, na situação sem política activa, as linhas no núcleo serem de 10 Mbps, e, na situação com política activa, serem no máximo de 30 Mbps. Justifica-se permitir um valor máximo de largura de

banda, no caso de largura de banda dinâmica, superior à largura de banda do caso de largura de banda fixa, pois a largura de banda libertada por alguns serviços pode ser utilizada por outros. Como se viu anteriormente, é possível libertar largura de banda mantendo a qualidade de serviço dentro de limites aceitáveis. Verifica-se que também é possível aceitar mais utilizadores, mantendo a qualidade de serviço dentro de limites aceitáveis.

É claro que o lucro obtido com o transporte de maior tráfego em EF e AF, e a economia de custos obtida com a libertação de largura de banda, proporcionada ao operador da rede por esta política activa, têm implicações nos atrasos sofridos pelos pacotes, visto que a carga é maior e os tempos de transmissão dos pacotes aumentam, quando a largura de banda é reduzida. A segunda regra de actuação, que age com base na informação de cumprimento dos SLAs de atraso proporcionada por outras políticas activas, permitirá que este aumento do atraso se mantenha dentro dos limites pretendidos.

Atendendo às limitações impostas pelo corretor de largura de banda, para valores elevados de carga oferecida à rede, o número de ligações que vêm o seu tráfego ser transportado numa classe inferior à pretendida torna-se significativo. Como esta degradação da classe de tráfego é indesejável para os utilizadores, serão utilizadas outras políticas activas para tentar minorar este problema.

A figura 5.25 ilustra o débito médio (em bit/segundo) obtido pelas ligações nas diferentes classes de tráfego com e sem a utilização desta política activa. Verifica-se que o débito diminui à medida que a carga aumenta, visto que, havendo mais tráfego para transportar, os atrasos nas filas de espera são maiores, reduzindo o débito das fontes TCP. Para as classes EF e AF, a carga é limitada pelos corretores de largura de banda, que se torna estável quando se atinge o limite máximo patente na figura 5.24, daí que o débito também se torne aproximadamente fixo a partir desse limite. No caso do tráfego AF, há uma diminuição muito acentuada quando a carga começa a aumentar, pois o tráfego fora de perfil, correspondendo ao tráfego acima de 64 Kbps, sendo de menor prioridade, sofre cada vez mais descartes. Assim, com a utilização desta política activa, o débito médio em AF vai aproximar-se mais do débito pedido ao corretor de largura de banda, que é de 64 Kbps. O débito médio em BE piora, pois reduzindo-se a largura de banda média no núcleo e sendo admitidas mais ligações em classes com maior prioridade, faz-se com que sobre menos largura de banda para a classe BE.

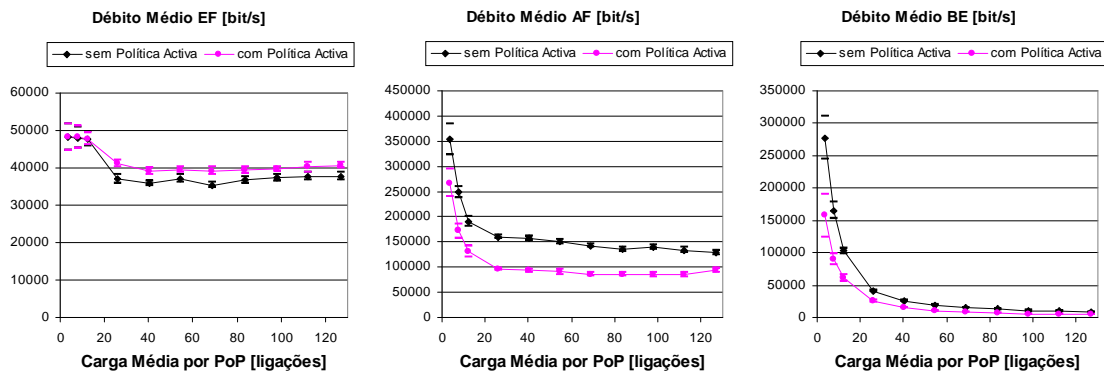


Figura 5.25: Comparação do débito com e sem a política activa de gestão de largura de banda.

A figura 5.26 ilustra o atraso médio (em segundos) sofrido pelos pacotes extremo a extremo, e a figura 5.27 o *jitter* médio (em segundos), obtidos para as diferentes classes de tráfego com e sem a utilização desta política activa. Verifica-se que o atraso médio obtido com a utilização da política activa é sempre superior, visto que ao diminuir-se a largura de banda nas linhas do núcleo está-se a aumentar os tempos de transmissão, e por consequência os atrasos sofridos pelos pacotes. Relativamente ao *jitter*, verifica-se que é mais elevado na situação com a política activa, mas, na classe EF, para valores elevados da carga, as diferenças tornam-se menos significativas.

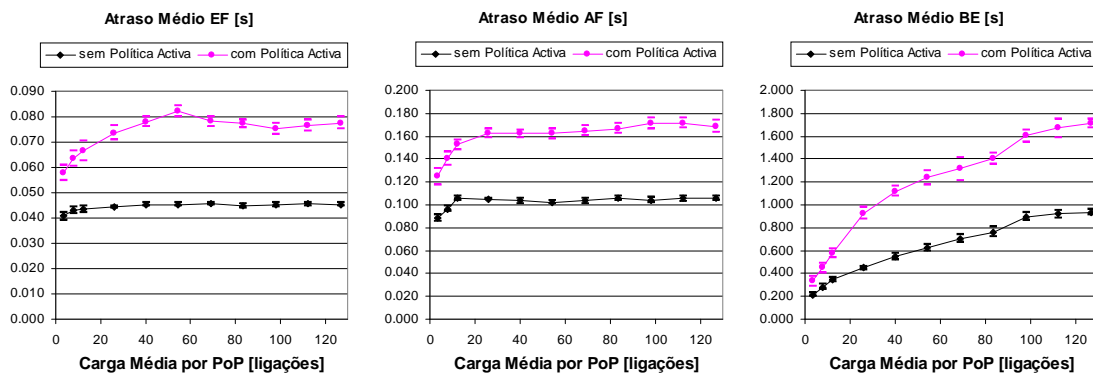


Figura 5.26: Comparação do atraso com e sem a política activa de gestão de largura de banda.

Conclui-se que esta política activa permite aumentar o lucro do operador da rede ajustando dinamicamente a largura de banda das linhas do núcleo da rede. Este lucro é obtido de duas formas: se a largura de banda puder aumentar, podem admitir-se mais

utilizadores, pelo que se vai obter uma maior facturação; por outro lado, se a largura de banda puder diminuir, pode-se ter uma largura de banda média mais baixa, libertando recursos, e assim reduzir os custos de operação da rede. Em qualquer dos casos, as modificações da largura de banda são feitas de forma a manter a qualidade de serviço oferecida aos utilizadores dentro dos limites pretendidos.

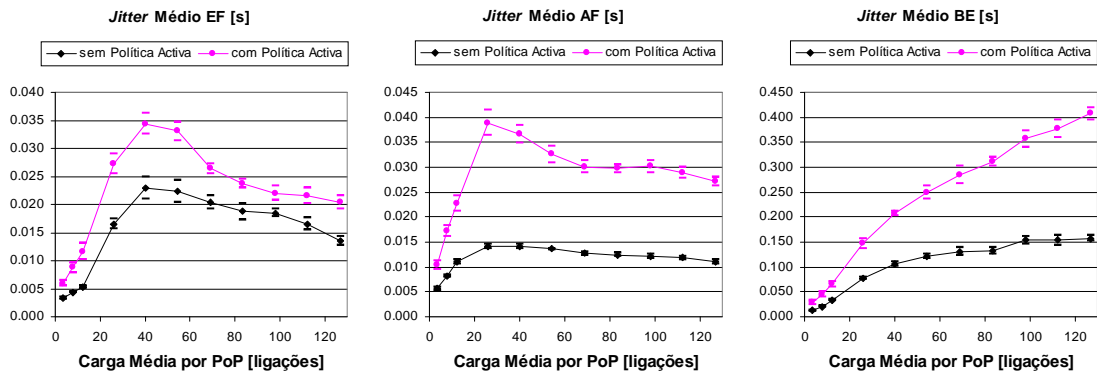


Figura 5.27: Comparação do *jitter* com e sem a política activa de gestão de largura de banda.

No entanto, caso se reduza a largura de banda, prejudica-se a qualidade de serviço oferecida aos utilizadores. Uma vez que esta política activa é local a cada linha do núcleo, observando-a de forma independente do resto da rede, não lhe é possível determinar se um SLA está a ser cumprido ou não. Para isso, terá de se ter uma política activa funcionando num nível acima a verificar os parâmetros de qualidade de serviço extremo a extremo, e a fornecer indicações a esta política activa de forma a economizar custos e manter a qualidade dentro dos limites impostos pelo SLA.

### 5.4.3 Política Activa de Gestão de Admissão de Ligações

As políticas activas de gestão de admissão de ligações têm como objectivo fazer o controlo de admissão de novas ligações, de forma a minimizar as ligações recusadas, e a manter a qualidade de serviço pretendida para cada classe de tráfego. Para isso, admite-se, tal como descrito na subsecção 5.2.3.1, que na classe AF é possível admitir mais ligações do que o previsto pelo pior caso possível, visto que o tráfego de algumas ligações é unidireccional ou intermitente. Já para a classe EF mantém-se o funcionamento do corretor de largura de banda, visto que se pretende não prejudicar a qualidade de serviço desta classe de tráfego.

Estas políticas activas existem numa versão local a cada ponto de acesso à rede, e numa versão global que abrange toda a rede.

A versão local a cada PoP substitui o corretor de largura de banda local ao PoP, na parte respeitante à classe AF, tomando decisões sobre se podem ou não ser admitidas mais ligações na classe AF. Para isso, esta política activa monitoriza continuamente os descartes de pacotes AF dentro de perfil e o comprimento médio da fila de espera, para ter uma noção da carga que esta classe tem.

A versão global a toda a rede limita-se a receber notificações de sobrecarga e a suspender temporariamente a admissão de novas ligações na classe AF em toda a rede, até que os problemas de sobrecarga estejam resolvidos. Estas situações de sobrecarga normalmente só acontecem quando se impõem SLAs impossíveis de cumprir.

A versão local desta política calcula cinco parâmetros de qualidade de serviço:

$$QoS(descartes) = \text{número de descartes de pacotes dentro de perfil em AF} \quad (5.8)$$

$$QoS(fila) = \text{MAX ( comprimento médio da fila AF de entrada no PoP, comprimento médio da fila AF de saída do PoP)} \times 4 / \text{comprimento máximo da fila AF} \quad (5.9)$$

$$QoS(EF \rightarrow AF) = \text{número de ligações que pretendiam EF e não conseguiram} \quad (5.10)$$

$$QoS(AF \rightarrow BE) = \text{número de ligações que pretendiam AF e não conseguiram} \quad (5.11)$$

$$QoS(recusados) = \text{número de ligações que foram recusadas} \quad (5.12)$$

O primeiro parâmetro,  $QoS(descartes)$ , dá uma indicação do número de pacotes descartados. Só se contam os descartes de pacotes AF dentro de perfil, pois são os pacotes que não devem ser descartados pela rede. Um valor maior para este parâmetro de QoS representa uma situação pior. Pretende-se que este parâmetro seja o mais baixo possível.

O segundo parâmetro,  $QoS(fila)$ , dá uma indicação da ocupação média das filas de espera e do consequente atraso sofrido pelos pacotes. Calcula-se o máximo dos valores das filas de entrada e de saída para representar o sentido com maior tráfego. Utilizou-se o valor médio do comprimento, calculado de acordo com o descrito na secção 5.2.4.2, e não o próprio comprimento da fila de espera, pois pretende-se ter uma ideia da ocupação média das filas, e não da ocupação instantânea. Neste caso, o comprimento da fila AF inclui tanto pacotes dentro, como fora de perfil. Para as filas de espera de AF, configuraram-se os limiares de descarte tal como sugerido na secção 5.2.4.2, pelo que  $QoS(fila)$  vale 0.5 quando se atinge o limiar mínimo para pacotes fora de perfil, 1.0 quando se atinge o limiar máximo para pacotes fora de perfil, que coincide com o limiar mínimo para pacotes dentro de perfil, e 2.0 quando se atinge o limiar máximo para pacotes dentro de perfil. Como facilmente se nota, um valor maior para este parâmetro de QoS corresponde a maiores atrasos verificados pelos pacotes, o que representa uma situação pior. A constante da fórmula foi escolhida, tal como na política activa de gestão de largura de banda das linhas, para dar o valor 1.0 no limite da situação normal, em que ainda não há descarte de pacotes AF dentro de perfil.

O terceiro parâmetro,  $QoS(EF \rightarrow AF)$ , dá uma indicação do número de ligações que viram a sua classe de tráfego ser degradada de EF para AF. Neste caso, não só os clientes são prejudicados por não obterem a classe pretendida, como o operador de rede vê os seus lucros diminuídos. Assim, pretende-se que este parâmetro seja o menor possível.

O quarto parâmetro,  $QoS(AF \rightarrow BE)$ , dá uma indicação do número de ligações que viram a sua classe de tráfego ser degradada de AF para BE. Para este caso, podem-se tecer considerações semelhantes às do caso anterior.

O quinto parâmetro,  $QoS(recusados)$ , dá uma indicação do número de ligações que foram recusadas por ter sido excedido o número máximo de ligações simultâneas.

O número máximo de ligações admitidas num dado PoP na classe AF é ajustado entre dois valores limites de acordo com a regra `threshold_decrement` aplicada ao valor de  $QoS(descartes) + QoS(fila)$  a intervalos de um segundo. O funcionamento desta regra está ilustrado na figura 5.28, com indicação dos valores utilizados para os diversos parâmetros. Há três zonas de funcionamento. Na zona em que o valor deste parâmetro de QoS de gestão é baixo, correspondente a valores inferiores a 0.25, permite-se a entrada de mais uma ligação do que há actualmente no PoP, desde que não

se atinja o valor máximo possível. Este valor máximo foi configurado suficientemente alto para que a limitação advenha antes da degradação da QoS. Na zona em que a QoS é normal, correspondente a valores entre 0.25 e 0.75, não se permite a entrada de mais ligações do que já há no PoP. Na zona em que a QoS é alta, correspondente a valores superiores a 0.75, baixa-se o número máximo de ligações de uma unidade, mantendo-se este valor sempre pelo menos em 6, tal como calculado anteriormente na tabela 5.8. Isto não reduz o número de ligações no PoP, mas se elas terminarem, não serão substituídas por outras novas, pelo que o número de ligações tenderá a diminuir.

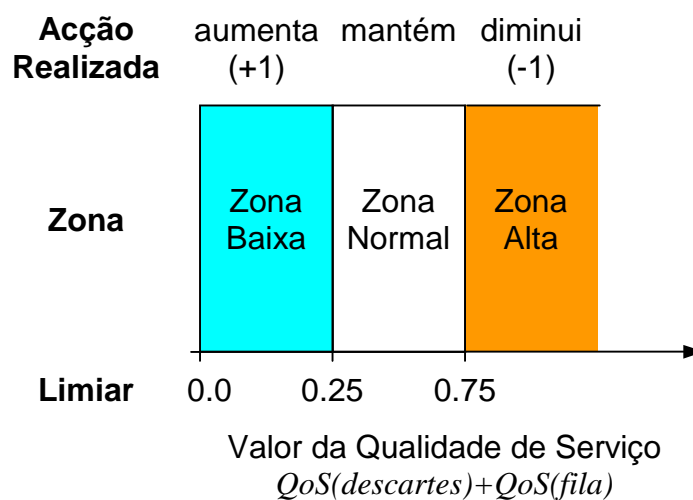


Figura 5.28: Funcionamento da regra de ajuste do número máximo de ligações em AF.

O funcionamento de uma primeira versão desta política activa foi descrito em [Pereira 00a] e [Pereira 00c].

Uma segunda regra reduz o número máximo de ligações na classe AF, mantendo o mesmo valor mínimo, caso o  $SLA(atrasoAF)$  esteja a ser violado. Esta regra só actua por indicação de uma política num nível superior, pelo que nos resultados desta secção esta regra nunca é activada, visto que se está a analisar cada política activa em separado.

A figura 5.29 compara o número médio de ligações nas várias classes de tráfego por PoP com a política activa de gestão de admissão de ligações e os correspondentes valores sem essa política activa, caso em que se tem o corretor de largura de banda a funcionar normalmente. Como esta política activa não afecta a admissão de ligações EF, o número de ligações nesta classe é exactamente igual nas duas situações. Relativamente à classe AF, verifica-se que, na situação em que se usa o corretor de



largura de banda, o número médio de ligações AF por PoP não ultrapassa o valor de 4.1 calculado anteriormente na tabela 5.8, enquanto que, se a política activa for utilizada, são permitidas muito mais ligações nesta classe de tráfego, o que vai aumentar os lucros do operador da rede. Como consequência, o número de ligações na classe BE é mais baixo, visto que menos ligações vêm a sua classe de tráfego ser degradada para BE.

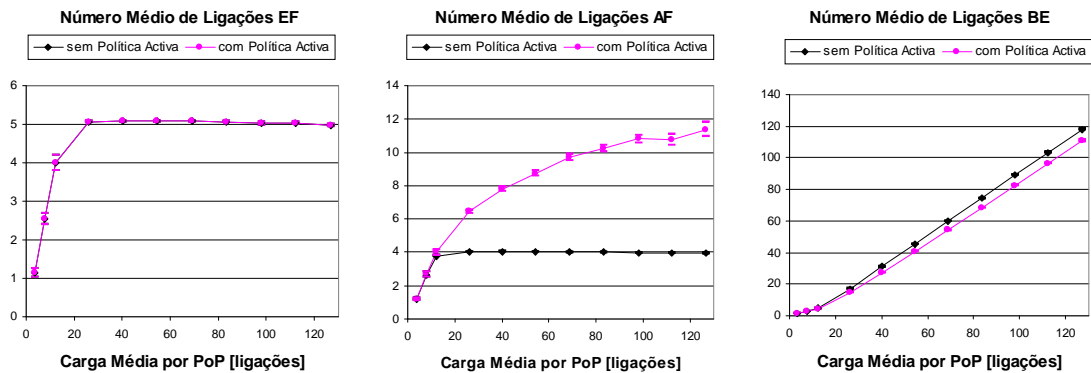


Figura 5.29: Comparação do número de ligações com e sem a política activa de gestão de admissão de ligações.

A figura 5.30 ilustra o débito médio (em bit/segundo) obtido pelas ligações nas diferentes classes de tráfego com e sem a utilização desta política activa. Verifica-se que o débito cai à medida que a carga aumenta, visto que havendo mais tráfego para transportar, os atrasos nas filas de espera são maiores, diminuindo o débito das fontes TCP. Para a classe EF, as diferenças entre as duas situações são mínimas, visto a carga ser limitada pelos corretores de largura de banda. No caso do tráfego AF, há uma diminuição muito acentuada quando a carga começa a aumentar, pois o tráfego fora de perfil, correspondendo ao tráfego acima de 64 Kbps, sendo de menor prioridade, sofre cada vez mais descartes à medida que são admitidos mais utilizadores. Verificou-se que nunca chegou a haver descarte de pacotes AF dentro de perfil, tal como se pretendia, embora haja essa possibilidade em situações de carga elevada, visto que as ligações, depois de admitidas, já não podem ser excluídas do sistema. Assim, o preço a pagar pela admissão de mais utilizadores em AF é o seu débito médio aproximar-se mais do débito pedido ao corretor de largura de banda, que é de 64 Kbps, o que não causa problemas ao operador da rede. O débito médio em BE piora ligeiramente, pois sendo admitidas mais ligações em classes com maior prioridade, sobra menos largura de banda para a classe BE.

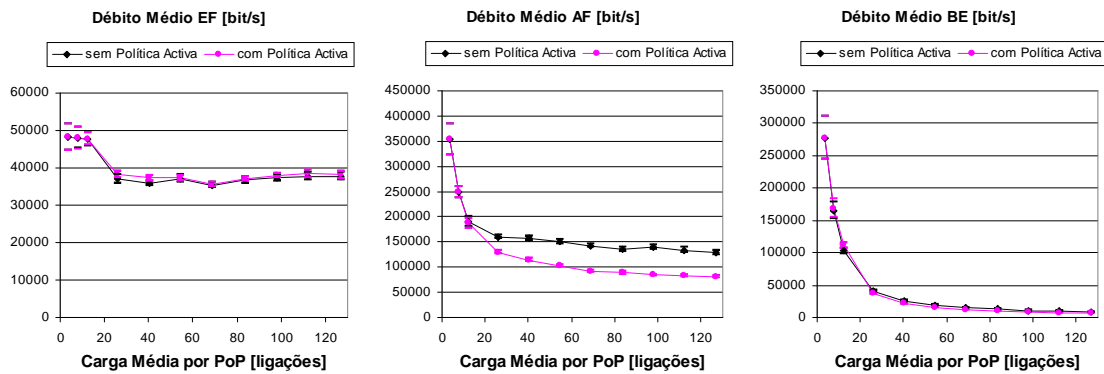


Figura 5.30: Comparação do débito com e sem a política activa de gestão de admissão de ligações.

A figura 5.31 ilustra o atraso médio (em segundos) sofrido pelos pacotes extremo a extremo, e a figura 5.32 o *jitter* médio (em segundos), obtidos para as diferentes classes de tráfego com e sem a utilização desta política activa. Verifica-se que o atraso médio obtido com a utilização da política activa é superior para as situações em que se admitiu um maior número de ligações, visto que se está a aumentar o tráfego, e, por isso, os atrasos nas filas de espera. As diferenças no atraso em EF para as duas situações são mínimas, visto o número de ligações ser igual e esta classe de tráfego ter maior prioridade. Por outro lado, em AF e BE, a mesma largura de banda é partilhada por um maior número de ligações, pelo que os atrasos pioram para cargas elevadas. Relativamente ao *jitter*, verifica-se que é aproximadamente igual nas situações de cargas baixas, piorando um pouco quando a carga aumenta, especialmente na classe BE.

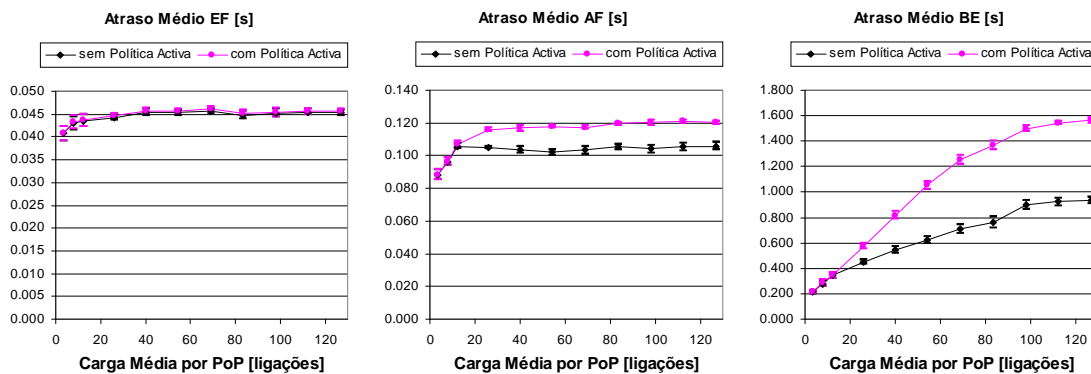


Figura 5.31: Comparação do atraso com e sem a política activa de gestão de admissão de ligações.

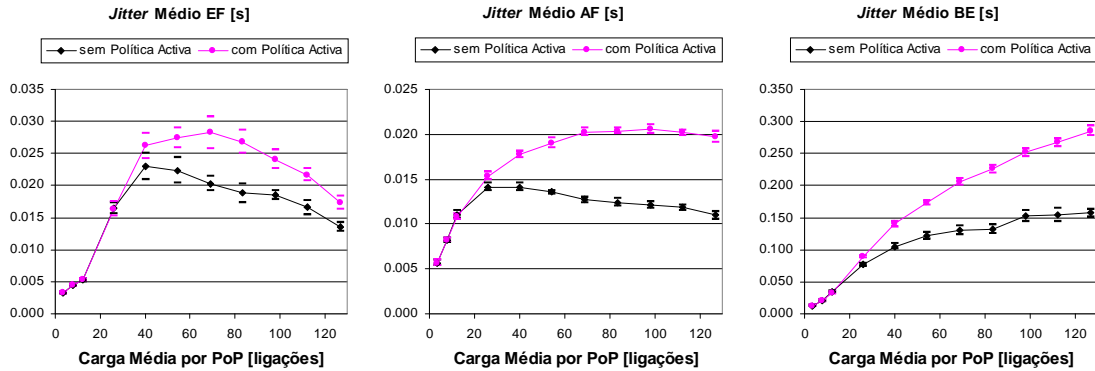


Figura 5.32: Comparação do *jitter* com e sem a política activa de gestão de admissão de ligações.

Conclui-se que esta política activa permite aumentar o lucro do operador da rede ajustando dinamicamente a admissão de ligações na classe AF, permitindo-se em média aceitar mais ligações nesta classe, sem afectar a classe EF nem degradar significativamente a qualidade de serviço obtida pelos utilizadores em AF. As principais diferenças verificadas com a admissão de mais ligações em AF são: o débito obtido por cada ligação não subir tanto acima do valor negociado com o corretor de largura de banda, quando a carga é grande; e o atraso e o *jitter* sofridos pelos pacotes aumentarem ligeiramente devido ao aumento da carga. Controlando o número máximo de ligações admitidas na classe AF, será possível manter a qualidade de serviço dentro dos limites impostos por um SLA.

#### 5.4.4 Política Activa de Gestão de Linhas Extra

As políticas activas de gestão de linhas extra têm como objectivo gerir a activação de linhas extra quando há uma falha numa linha, ou uma sobrecarga que cause uma degradação local da qualidade de serviço.

O algoritmo implementado nestas políticas activas é um algoritmo hierárquico. É composto por um algoritmo intra-região, com a responsabilidade de gerir as linhas extra dentro dessa região, e um algoritmo inter-regiões, com a responsabilidade de gerir as linhas extra entre as várias regiões da rede.

A topologia da rede, apresentada anteriormente na figura 5.12, foi escolhida de forma a minimizar o número de linhas e o número de *routers* com capacidade de estabelecer linhas extra. Além disso, pretendeu-se ainda garantir que se mantém a

conectividade em todos os nós da rede, mesmo com a falha de uma linha em cada região e de uma linha entre regiões.

Para garantir estas condições, a rede tem um caminho dentro de cada região a ligar todos os nós. Além disso, foram colocados dois *routers* com capacidade de estabelecer linhas extra em cada região nos nós com apenas uma linha, visto que são os que mais facilmente ficam isolados no caso de falha de linhas. Desta forma, pode-se fechar um anel dentro de cada região, permitindo-se também aumentar a capacidade de transporte de tráfego dentro de cada região, se tal for necessário. Finalmente, cada região é ligada a outra região através de uma linha entre nós já com duas linhas, os menos sujeitos a falhas.

Há ligeiras diferenças no funcionamento do algoritmo consoante se esteja na presença de falhas de linhas, ou de sobrecarga, pelo que se começa por descrever a situação de falha de linhas. O funcionamento destas políticas apenas na situação de falhas foi descrito em [Pereira 00c].

Quando há uma falha de uma linha, é gerada uma notificação de falha, em ambos os extremos da linha, para que a falha seja rapidamente resolvida e para que se possam tomar medidas com vista a evitar a sobrecarga da rede. Tais medidas incluem a activação de linhas extra, podendo ainda incluir a modificação temporária do mapeamento do tráfego das aplicações nas classes de tráfego, ou mesmo o descarte de tráfego de baixa prioridade, consoante as decisões tomadas por outras políticas activas.

A informação da falha de uma linha propaga-se na rede, a partir de ambos os extremos da linha em falha, até chegar aos nós com capacidade de estabelecer linhas extra dessa região, que correm os algoritmos de gestão de linhas extra.

Os nós com capacidade de estabelecer linhas extra conhecem: todos os nós para os quais podem estabelecer linha extra e a distância a que está cada um; a topologia da rede da sua região e as linhas para regiões adjacentes; e ainda quais as ligações em baixo em cada momento que envolvem a sua região, através das informações de falhas recebidas.

Os algoritmos de gestão de linhas extra procuram a melhor linha alternativa a ligar na topologia da rede que conhecem. Se tiver falhado uma linha dentro de uma única região, o algoritmo intra-região tenta ligar a linha alternativa dessa região, o que resolve o problema. Se essa linha já estivesse ligada, ou a falha envolve duas regiões, o problema passa para o algoritmo inter-regiões. O algoritmo inter-regiões verifica se há

perda de conectividade em algum nó da rede, pois poderia haver alguma linha extra ligada que ainda não tivesse sido desligada e que assim resolvesse o problema. Caso haja perda de conectividade, é ligada a linha extra mais próxima de um lado para o outro da linha em falha, de forma a repor a conectividade. Como os vários nós com capacidade de estabelecer ligações extra correm o mesmo algoritmo, chegam todos à mesma decisão de qual a linha extra a ligar. No entanto, no caso de falhas múltiplas, nem sempre é possível manter a conectividade (e.g. se falharem as linhas 0-4 e 4-2, o nó 4 fica isolado).

As linhas extra ligadas são automaticamente desligadas quando deixam de ser necessárias. No entanto, no caso de falhas múltiplas, isso pode não corresponder à resolução da falha que deu origem à activação da linha. Assim, optou-se por só desactivar uma linha alternativa quando todas as falhas nas regiões abrangidas por essa linha alternativa estiverem resolvidas.

A tabela 5.9 mostra a taxa de sucesso, destas políticas activas, na recuperação de falhas. Os intervalos de confiança a 95% foram calculados com base no exposto no apêndice C, tendo sido realizadas 12 simulações de duração igual a 600 vezes o tempo médio entre falhas. A política local a cada região resolve aproximadamente metade das falhas. A política global resolve o problema em 18.7% dos casos. 17.0% dos casos correspondem a situações em que já tinha sido activada uma linha extra, e em que não foi necessário tomar nenhuma acção. Finalmente, 13.7% das falhas correspondem a falhas múltiplas que não podem ser resolvidas com a topologia escolhida para a rede, resultando em que pelo menos um nó fique desligado do resto da rede enquanto durar a falha.

|   | <b>Percentagem de Casos</b> | <b>Intervalo de Confiança a 95%</b> |
|---|-----------------------------|-------------------------------------|
| <b>Falhas Totais</b>                          | 100.0%                      |                                     |
| <b>Resolvidas pela política Intra-Região</b>  | 50.6%                       | ±1.2%                               |
| <b>Resolvidas pela política Inter-Regiões</b> | 18.7%                       | ±0.9%                               |
| <b>Sem Problema</b>                           | 17.0%                       | ±1.0%                               |
| <b>Não Resolvidas</b>                         | 13.7%                       | ±0.8%                               |

Tabela 5.9: Taxa de sucesso da política activa de gestão de linhas extra na recuperação de falhas.

Estes resultados mostram a vantagem de utilizar um algoritmo hierárquico, visto que os algoritmos internos a cada região resolvem cerca de metade dos problemas, mas o algoritmo de topo é necessário para resolver problemas que envolvam regiões diferentes.

Relativamente à sobrecarga de linhas, podem distinguir-se duas situações. Na situação em que há sobrecarga de uma linha bem determinada, reage-se como na situação de falha, procurando a linha extra mais próxima para servir de alternativa. Por outro lado, na situação de sobrecarga geral da rede, detectada pela violação de algum SLA, ligam-se as linhas que conduzem a uma maior redução dos percursos médios na rede, admitindo todos os destinos equiprováveis e todos os nós com o mesmo tráfego, começando pelas linhas internas a cada região. A tabela 5.10 mostra o comprimento médio dos percursos no núcleo da rede com diferentes possibilidades de linhas extra ligadas. Uma vez que o benefício de ligar mais que as 5 primeiras linhas extra já é relativamente reduzido, optou-se por utilizar apenas estas linhas que estão ilustradas na figura 5.33, com indicação da ordem por que são ligadas.

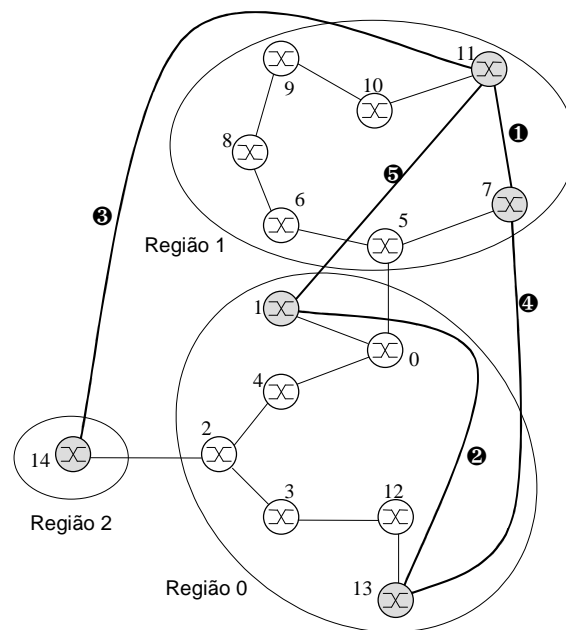


Figura 5.33: Topologia do núcleo da rede com indicação das linhas extra utilizadas.

| <b>Linhas Extra Ligadas</b>   | <b>Percurso Médio</b> |
|-------------------------------|-----------------------|
| Nenhuma                       | 4.267                 |
| 7-11                          | 3.829                 |
| 7-11, 1-13                    | 3.429                 |
| 7-11, 1-13, 11-14             | 2.990                 |
| 7-11, 1-13, 11-14, 7-13       | 2.724                 |
| 7-11, 1-13, 11-14, 7-13, 1-11 | 2.629                 |
| Todas as 10 possíveis         | 2.457                 |

Tabela 5.10: Comprimento médio dos percursos no núcleo da rede com diferentes possibilidades de linhas extra ligadas.

A política activa de gestão de linhas extra utiliza um parâmetro de qualidade de serviço calculado na política activa de gestão de SLAs:

$$QoS(atraso) = MAX \left( \frac{atrasoEF}{SLA(atrasoEF)}, \frac{atrasoAF}{SLA(atrasoAF)} \right) \quad (5.13)$$

que representa o maior dos atrasos das classes EF e AF relativamente ao especificado no SLA. Este parâmetro é calculado a intervalos de 1 segundo e 5 segundos. Pretende-se que este parâmetro de QoS seja menor ou igual a 1, significando que se está a cumprir os SLAs.

Esta política activa actua ligando mais uma linha extra na ordem indicada na tabela 5.10, quando a  $QoS(atraso)$  calculada a intervalos de 1 segundo excede 1.0, conforme ilustrado na figura 5.34. E actua também desligando uma linha extra na ordem inversa, quando tanto a  $QoS(atraso)$  calculada a intervalos de 1 segundo, como a calculada a intervalos de 5 segundos, estiverem abaixo de 0.9.

Para desligar uma linha, primeiro o seu custo para o algoritmo de encaminhamento é aumentado significativamente, de forma a que a linha deixe de ser utilizada para encaminhar tráfego. Só depois, caso as condições se mantenham, é que a linha é de facto desligada, permitindo-se assim que os pacotes que já estivessem nas filas de espera sejam transmitidos, sem qualquer descarte.

Para testar o funcionamento desta política activa, utilizou-se para o  $SLA(atrasoEF)$  o valor de 30 ms e para  $SLA(atrasoAF)$  o valor de 55 ms. Estes valores são suficientemente baixos para que a política activa actue rapidamente, ligando as cinco linhas extra em cinco segundos.

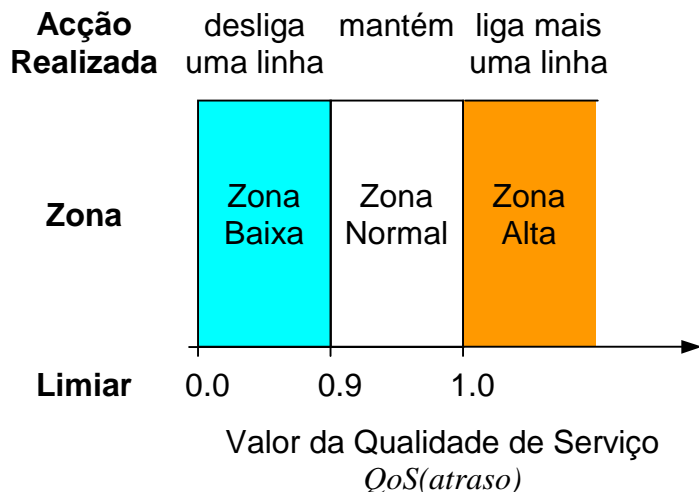


Figura 5.34: Funcionamento da política de activação de linhas extra.

A figura 5.35 ilustra o atraso médio (em segundos) sofrido pelos pacotes extremo a extremo, com e sem a política activa de gestão de linhas extra para as diferentes classes de tráfego. Verifica-se que o benefício médio na classe EF é de 8.4 ms, na classe AF é de 4.7 ms, e na classe BE é aproximadamente nulo. Estas diferenças são devidas ao facto de a classe EF ser a menos carregada, pelo que os seus pacotes quase sempre têm créditos para serem transmitidos, pelo que o benefício obtido pela redução do comprimento do percurso médio se traduz directamente numa redução no atraso sofrido pelos pacotes. A redução média esperada será de:

$$\begin{aligned} & \text{redução no comprimento do percurso} \times (T_{\text{transmissão}} + T_{\text{propagação}}) = \\ & = (4.267 - 2.629) \cdot \left( \frac{1000 * 8}{10 * 10^6} + 0.005 \right) = 9.5 \text{ ms.} \end{aligned} \quad (5.14)$$

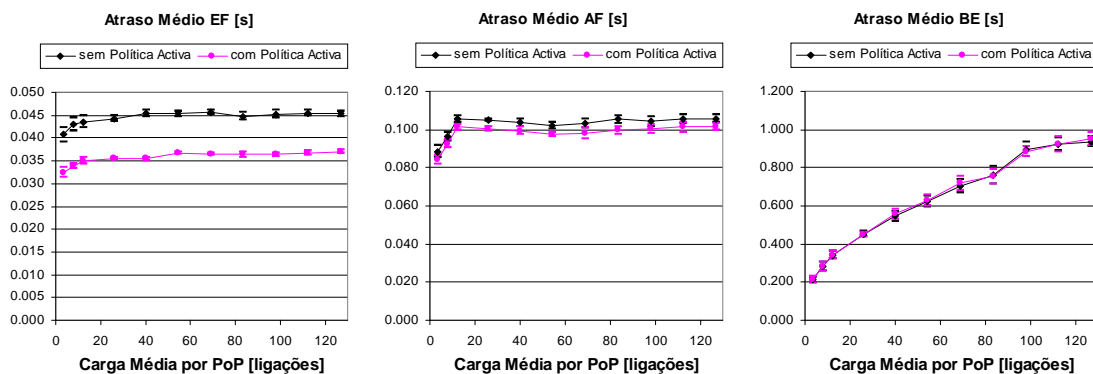


Figura 5.35: Comparação do atraso com e sem a política activa de gestão de linhas extra.



Para as outras classes de tráfego é mais difícil de prever o efeito, pois uma redução do atraso extremo a extremo resulta num aumento do débito, que contraria a redução do atraso obtida.

Conclui-se que esta política activa permite reduzir significativamente os efeitos de perda de disponibilidade do serviço fornecido pela rede no caso de falha de linhas, ou reduzir os atrasos sofridos pelos pacotes, a custo de activar linhas extra.

#### 5.4.5 Política Activa de Gestão de SLAs

As políticas activas de gestão de SLAs têm como objectivo garantir que os SLAs são cumpridos. Para isso, monitorizam a qualidade de serviço obtida pelos utilizadores, e ajustam dinamicamente os parâmetros das políticas activas dos níveis inferiores.

Esta política activa deve ter módulos de monitorização nos vários pontos da rede onde devem ser feitas medidas dos parâmetros de QoS especificados nos SLAs, e um módulo central onde os dados são agrupados e as principais decisões de gestão tomadas.

A política activa de gestão de SLAs faz ajustes nas quatro políticas activas descritas anteriormente (nas subsecções 5.4.1 a 5.4.4), quando algum limite especificado num SLA é ultrapassado:

- Na política activa de gestão dos pesos das classes de tráfego, modificando o parâmetro *AFOutPercent*, e assim gerindo quais as linhas em que é feito o descarte de pacotes AF fora de perfil.
- Na política activa de gestão de largura de banda das linhas, aumentando a largura de banda das linhas.
- Na política activa de gestão de admissão de ligações, reduzindo o número máximo de utilizadores permitidos na classe AF.
- Na política activa de gestão de linhas extra, activando linhas extra.

Uma vez que só faz sentido ter esta política em funcionamento simultaneamente com as outras, só se apresentam os resultados obtidos na secção seguinte (5.5), terminando-se esta subsecção com a descrição de como o parâmetro *AFOutPercent* é modificado.

Relembre-se que o parâmetro *AFOutPercent* determina o descarte do tráfego AF fora de perfil quando o atraso dos pacotes nas filas de espera ultrapassa a correspondente fracção do SLA, permitindo descartar tráfego fora de perfil de forma

mais agressiva nas linhas mais congestionadas, sendo utilizado em situações em que o SLA é ultrapassado. O parâmetro *AFOutPercent* é ajustado entre 0.0 e 1.0, de acordo com a regra *threshold\_decrement* aplicada ao valor de *QoS(atrasoAF)* calculado a intervalos de um segundo, conforme ilustrado na figura 5.36. Na zona em que a QoS é alta, correspondente a valores superiores a 1.0, caso em que o limite do SLA está a ser ultrapassado, baixa-se *AFOutPercent* de 0.3 de cada vez, pelo que o tráfego AF fora de perfil será descartado em mais linhas. Na zona em que a QoS é normal, correspondente a valores entre 0.9 e 1.0, mantém-se o valor de *AFOutPercent*. Na zona em que a QoS é baixa, correspondente a valores inferiores a 0.9, sobe-se novamente *AFOutPercent* de 0.3 de cada vez.

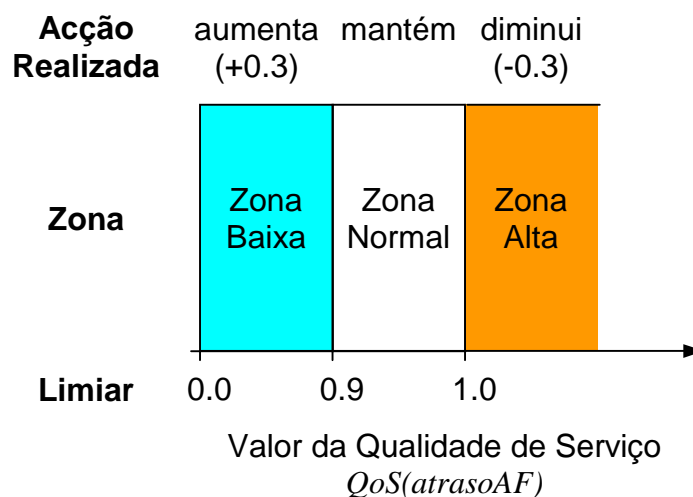


Figura 5.36: Funcionamento da regra de ajuste de *AFOutPercent*.

#### 5.4.6 Política Activa de Gestão do Mapeamento para PHBs

As políticas activas de gestão do mapeamento para PHBs têm como objectivo minimizar o número de ligações que obtêm uma classe de tráfego inferior à pretendida. Admite-se, tal como descrito na secção 5.2.2, que há regras que definem como mapear o tráfego dos utilizadores nas classes de tráfego. Tal mapeamento pode estar rigidamente ligado ao contrato que os clientes fazem com o fornecedor de serviço, ou dar a este alguma margem de manobra.

Admitiu-se que os clientes negociavam o mapeamento preferido do seu tráfego nas classes de tráfego, podendo o fornecedor de serviço Internet modificar este

mapeamento em situações críticas. Desta forma, o fornecedor de serviço pode, quando necessário, seleccionar as aplicações mais importantes dos clientes como candidatas a utilizar as classes de tráfego mais prioritárias, degradando a classe de tráfego das aplicações menos importantes. Esta selecção é modificada dinamicamente consoante as características de carga da rede. O objectivo é poder modificar a carga oferecida a cada classe de tráfego, de maneira a ser mais fácil assegurar a disponibilidade do serviço das classes de tráfego mais prioritárias.

Para atingir os objectivos propostos, definiram-se quatro regras de restrição do mapeamento do tráfego de cada aplicação do cliente nas classes de tráfego. Estas regras, descritas na tabela 5.11, impedem que certas aplicações do cliente utilizem as classes mais prioritárias. A regra 1 não faz qualquer restrição, permitindo que o cliente escolha a classe que entender. A regra 2 obriga a que o tráfego FTP vá na classe BE. A regra 3 adicionalmente impede que o tráfego HTTP vá na classe EF. Finalmente a regra 4 obriga o tráfego HTTP e FTP a ir na classe BE, e só permite tráfego CBR na classe EF.

| <b>Aplicação</b> | <b>Regra 1</b> | <b>Regra 2</b> | <b>Regra 3</b> | <b>Regra 4</b> |
|------------------|----------------|----------------|----------------|----------------|
| <b>Telnet</b>    | EF, AF, BE     | EF, AF, BE     | EF, AF, BE     | AF, BE         |
| <b>CBR</b>       | EF, AF, BE     | EF, AF, BE     | EF, AF, BE     | EF, AF, BE     |
| <b>OnOff</b>     | EF, AF, BE     | EF, AF, BE     | EF, AF, BE     | AF, BE         |
| <b>HTTP</b>      | EF, AF, BE     | EF, AF, BE     | AF, BE         | BE             |
| <b>FTP</b>       | EF, AF, BE     | BE             | BE             | BE             |

Tabela 5.11: Regras de restrição do mapeamento para PHBs.

Como se pode verificar, cada uma destas regras restringe mais do que a anterior o tráfego que pode utilizar as classes de tráfego mais prioritárias. Seria possível pensar em outras regras, chegando se necessário a descartar o tráfego FTP, para utilizar em situações muito graves, tais como a falha de linhas, mas considerou-se suficiente a utilização das quatro regras indicadas na tabela 5.11.

As políticas activas de gestão do mapeamento para PHBs existem localmente a cada ponto de acesso à rede, monitorizando os fluxos dos pedidos de estabelecimento de ligação nas classes EF e AF que obtiveram uma classe inferior à desejada. As políticas actuam mudando a regra de restrição do mapeamento no sentido 1 → 2 → 3 → 4, sempre que há uma ligação com a classe de tráfego degradada, e mudando a regra de restrição do mapeamento no sentido 4 → 3 → 2 → 1, sempre que há dez ligações das

classes EF e AF sem a classe de tráfego degradada. É utilizada a regra `count` para realizar a contagem do número de vezes que a ligação é, ou não, degradada. Novamente, para respeitar o algoritmo genérico de ajuste de parâmetros do sistema, descrito anteriormente na figura 3.10, basta uma ligação com a classe de tráfego degradada para se actuar, sendo o retorno à situação normal mais lento, sendo neste caso, necessárias 10 ligações sem a classe de tráfego degradada.

O funcionamento de uma primeira versão desta política activa foi descrito em [Pereira 00a] e [Pereira 00c].

Para analisar o efeito desta política activa, apresenta-se na tabela 5.12 a fracção do tempo que cada regra de restrição do mapeamento para PHBs é utilizada, para simulações com quatro valores de carga média por PoP. Verifica-se que, para carga 5, é sempre seleccionada a regra 1. Para carga 10, a maior parte do tempo é usada a regra 1, embora se note que a regra 4 já tem alguma utilização em alguns pontos de acesso em alturas críticas. Para a carga 15, a utilização da regra 1 é em média inferior, sendo frequente utilizar regras superiores à 1. Para o caso de carga 30, verifica-se que quase sempre é utilizada a regra 4, significando que a regra a utilizar muda rapidamente para a 4, onde se mantém. Não se representou os casos de carga superior, pois são casos em que a política tende ainda mais rapidamente para a regra 4.

| <b>Carga</b> | <b>Regra 1</b> | <b>Regra 2</b> | <b>Regra 3</b> | <b>Regra 4</b> |
|--------------|----------------|----------------|----------------|----------------|
| <b>5</b>     | 100.0%         | 0.0%           | 0.0%           | 0.0%           |
| <b>10</b>    | 77.6%          | 9.8%           | 3.8%           | 8.7%           |
| <b>15</b>    | 65.4%          | 16.6%          | 9.1%           | 8.9%           |
| <b>30</b>    | 0.4%           | 0.4%           | 3.2%           | 96.0%          |

Tabela 5.12: Fracção do tempo que cada regra de restrição do mapeamento para PHBs é usada.

A figura 5.37 ilustra a fracção do número de ligações que tiveram a classe de tráfego degradada de EF para AF, ou de AF para BE, nos casos com e sem a política activa de gestão do mapeamento para PHBs. Destes gráficos nota-se que, quando não se utiliza a política activa de gestão do mapeamento para PHBs, mesmo para situações com carga baixa, como a de 10 ligações por PoP em AF, já há uma certa fracção de ligações com a classe de tráfego degradada. Tal deve-se às situações em que muitas ligações têm

o mesmo nó de destino (ou de origem), pelo que o corretor de largura de banda tem de degradar a classe de algumas.

Analisando agora a situação com a utilização da política activa de gestão do mapeamento para PHBs, verifica-se que é possível reduzir muito a fracção de ligações EF que não consegue obter a classe de tráfego pretendida, de entre as seleccionadas para poder utilizar esta classe de tráfego. A redução na maior parte das situações é superior a 6 vezes. Já para o caso das ligações que pretendiam obter a classe AF, a redução não é tão grande, mas ainda assim é para menos de metade. Estas reduções são devidas a se ter admitido, conforme indicado na secção 5.3.3, que a classe de tráfego escolhida pelos clientes era aleatória, com todas as classes de tráfego equiprováveis em ambas as situações, mas na regra 4 só se permitir tráfego CBR em EF, obrigando ainda o tráfego FTP e HTTP a ir em BE.

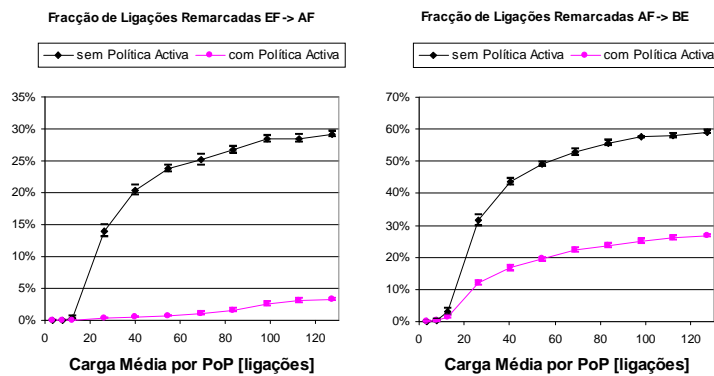


Figura 5.37: Comparação da fracção do número de ligações com a classe de tráfego degradada nas situações com e sem a política activa de gestão do mapeamento para PHBs.

A figura 5.38 ilustra o número médio de ligações nas várias classes de tráfego por PoP com e sem a política activa de gestão de mapeamento para PHBs. Da análise desta figura, conclui-se que a principal diferença é no número de ligações na classe EF que, por efeito da política activa, baixa, mantendo-se mais afastado da saturação. Este efeito nota-se a partir de uma carga de 30 ligações por PoP, em que a política activa já actuou, utilizando na maior parte dos casos a regra de restrição 4, pelo que o número de ligações a ser seleccionadas para tentar obter esta classe de tráfego é inferior, sendo menos recusadas, e melhorando a disponibilidade do serviço obtida.

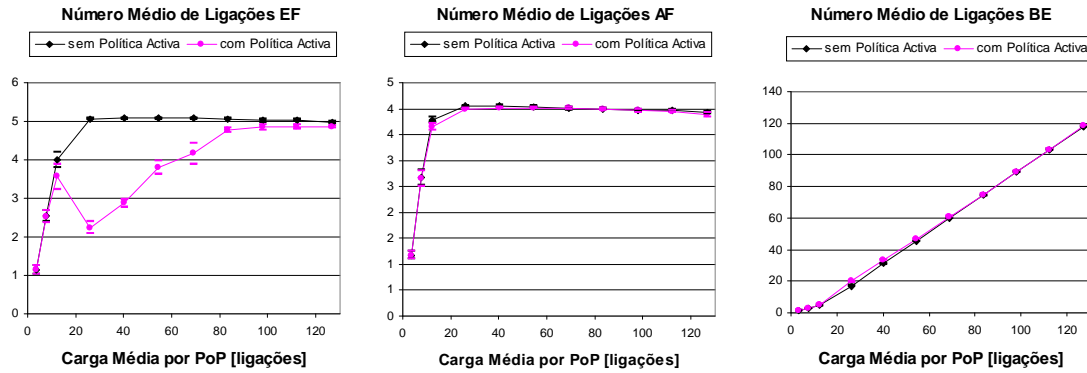


Figura 5.38: Comparação do número de ligações nas situações com e sem a política activa de gestão do mapeamento para PHBs.

A figura 5.39 ilustra o débito médio (em bit/segundo) obtido nas situações com e sem a política activa de gestão do mapeamento para PHBs. Verifica-se que o débito da classe EF aumenta quando a política activa começa a actuar. Tal é explicado por haver um aumento da percentagem de ligações CBR, que são de 64 Kbps, que usam esta classe, à medida que a política activa aumenta as restrições às aplicações que podem utilizar a classe EF. Uma vez que as aplicações que geram dados a ritmos mais elevados, FTP e HTTP, passam a utilizar a classe BE, é natural que haja uma diminuição do débito médio em AF, e em consequência sobrem mais recursos desta classe, pelo que o débito médio em BE aumenta.

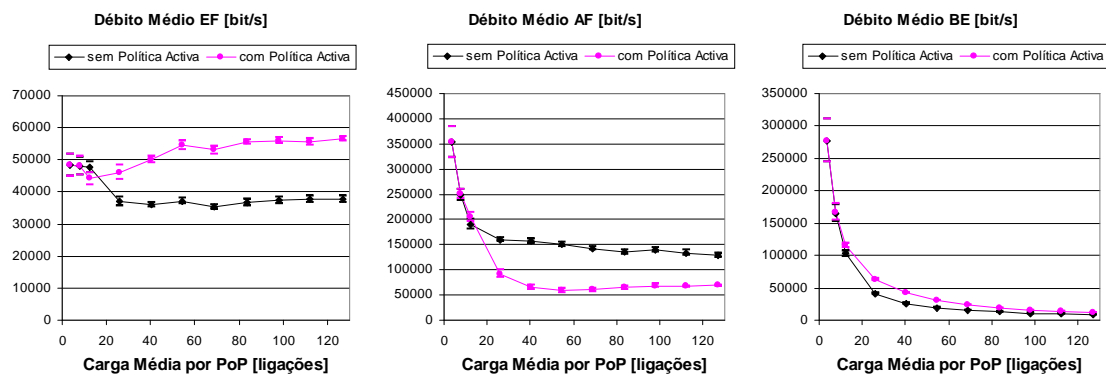


Figura 5.39: Comparação do débito nas situações com e sem a política activa de gestão do mapeamento para PHBs.

A figura 5.40 ilustra o atraso médio (em segundos) sofrido pelos pacotes extremo a extremo, e a figura 5.41 ilustra o *jitter* médio (em segundos), obtidos para as diferentes classes de tráfego com e sem a utilização desta política activa. Aqui, a

explicação das diferenças é semelhante à anterior. Ficando quase só tráfego CBR na classe EF, o *jitter* baixa significativamente, mantendo-se o atraso, visto esta classe ter em qualquer dos casos uma carga muito baixa. Como as aplicações que geram dados a ritmos mais elevados passam a utilizar a classe BE, a classe AF fica menos carregada, pelo que o seu atraso e *jitter* médios baixam. Finalmente, a classe BE, embora fique mais carregada, tem mais recursos disponíveis que sobram das outras classes, verificando-se que o nível de carga médio baixa, pelo que o atraso e o *jitter* baixam.

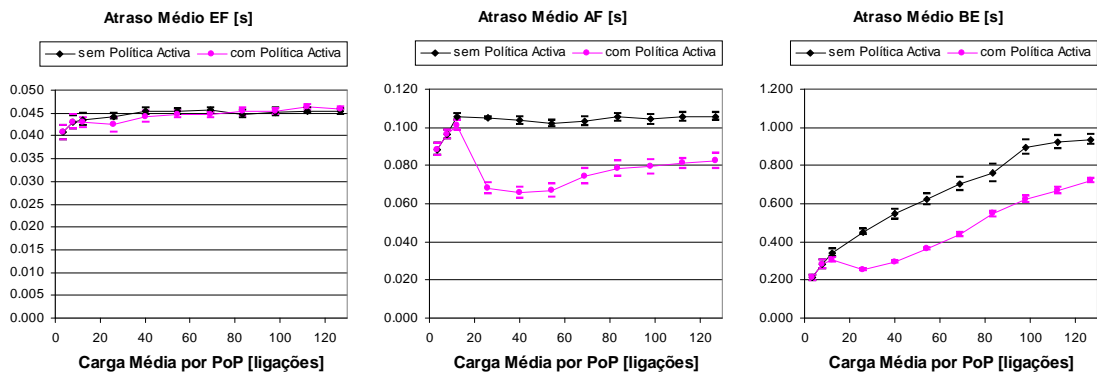


Figura 5.40: Comparação do atraso nas situações com e sem a política activa de gestão do mapeamento para PHBs.

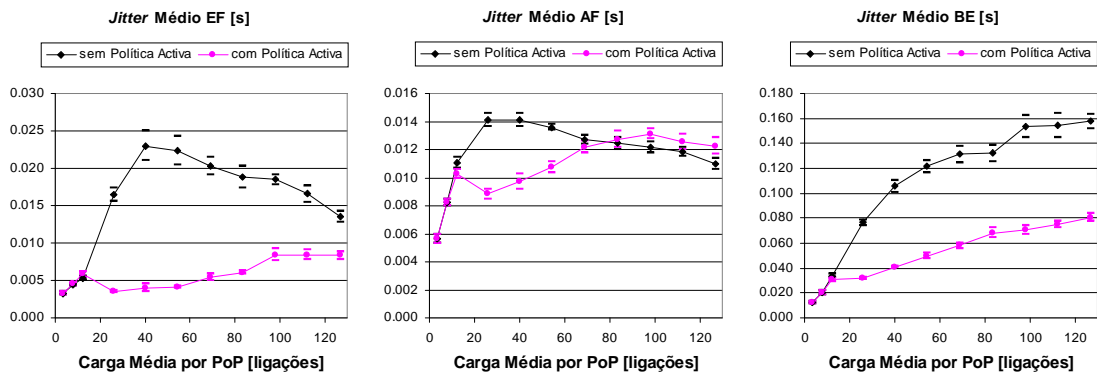


Figura 5.41: Comparação do *jitter* nas situações com e sem a política activa de gestão do mapeamento para PHBs.

Conclui-se que esta política é eficaz no aumento da disponibilidade dos serviços oferecidos pelo operador da rede. Isso é conseguido à custa de modificar os mapeamentos das aplicações dos utilizadores nas classes de tráfego, seleccionando-se apenas as aplicações mais importantes para utilizar as classes mais prioritárias.

## 5.5 FUNCIONAMENTO SIMULTÂNEO DAS VÁRIAS POLÍTICAS ACTIVAS

Nesta secção descrevem-se os resultados obtidos com todas as políticas activas em funcionamento simultâneo, sem falhas de linhas. Analisa-se primeiro a situação sem planeamento, e depois com uma política activa de planeamento.

### 5.5.1 Sem Política Activa de Planeamento

Começou-se por utilizar os valores de 60 ms para o  $SLA(atrasoEF)$  e 120 ms para o  $SLA(atrasoAF)$ , visto serem valores que é possível cumprir. Pretende-se, assim, que os atrasos médios sejam inferiores a estes valores limites para as classes EF e AF. A figura 5.42 ilustra os atrasos médios (em segundos) sofridos pelos pacotes extremo a extremo, para as diferentes classes de tráfego, sem a utilização das políticas activas e com as várias políticas activas a funcionar simultaneamente.

Verifica-se que os SLAs são ambos cumpridos, sendo a folga razoável no caso da classe AF, o que denota algum desaproveitamento de recursos. Este facto mostra como é difícil fazer cumprir com exactidão um SLA, visto que as políticas activas reagem quase todas com base em limiares fixos, correspondendo a situações próximas dos limites permitidos pelos SLAs. Assim, para fazer cumprir com exactidão os SLAs, será necessário recorrer a políticas activas de planeamento, que analisando os parâmetros com prazos mais largos, podem ajustar os limiares utilizados, de forma a melhorar a eficiência de utilização dos recursos.

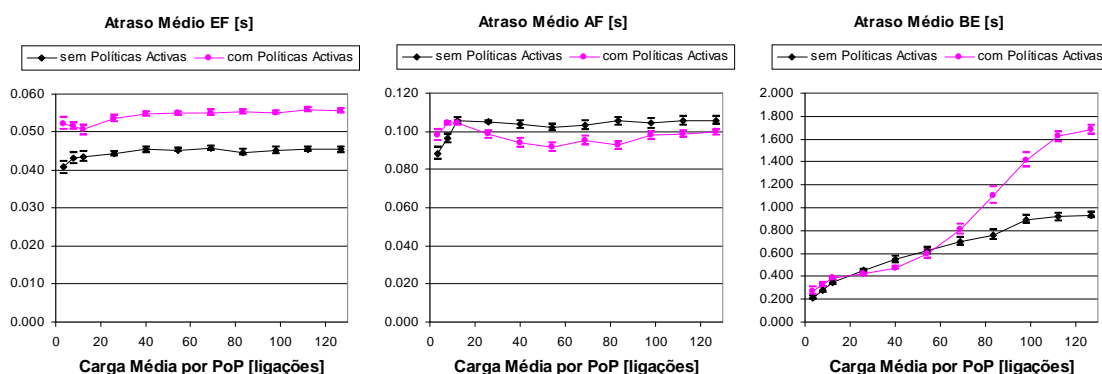


Figura 5.42: Comparação do atraso nas situações com e sem as políticas activas de gestão.



A figura 5.43 ilustra, à esquerda, a largura de banda média (em bit/segundo) nas linhas do núcleo da rede com e sem a utilização conjunta das políticas activas. À direita ilustra-se a largura de banda que foi efectivamente utilizada. A situação com utilização das políticas activas pode ter até 19 linhas activas no núcleo, se forem activadas linhas extra, enquanto na situação sem políticas activas estão activas apenas 14 linhas. Os dados apresentados neste gráfico para o caso de utilização das políticas activas já foram corrigidos por um factor de 19/14, para tornar a comparação válida.

Destes dados, conclui-se que é possível ao operador da rede oferecer aos utilizadores a qualidade de serviço requerida, reservando uma largura de banda mínima, e libertando, desta forma, recursos para outras utilizações. Este efeito é principalmente devido à política activa de gestão da largura de banda das linhas do núcleo da rede.

A libertação de largura de banda só é possível se toda a gestão for fiável e funcionar devidamente, pois de outro modo não se tem o controlo de quanta largura de banda está a ser utilizada, nem perspectivas da evolução do sistema, necessárias para actividades de planeamento. Com o uso das políticas activas é possível reutilizar os recursos excedentes de largura de banda.

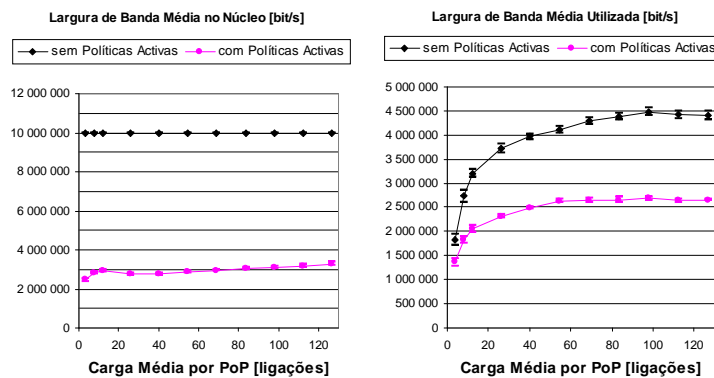


Figura 5.43: Comparação da largura de banda média e da largura de banda média utilizada nas situações com e sem as políticas activas de gestão.

A figura 5.44 ilustra o número médio de ligações com e sem a utilização das políticas activas. Verifica-se que, com a utilização das políticas activas, é possível transportar mais ligações na classe EF, e muito mais ligações na classe AF, do que na situação sem políticas activas. A grande diferença existente em AF é devida principalmente à política activa de gestão de admissão de ligações.

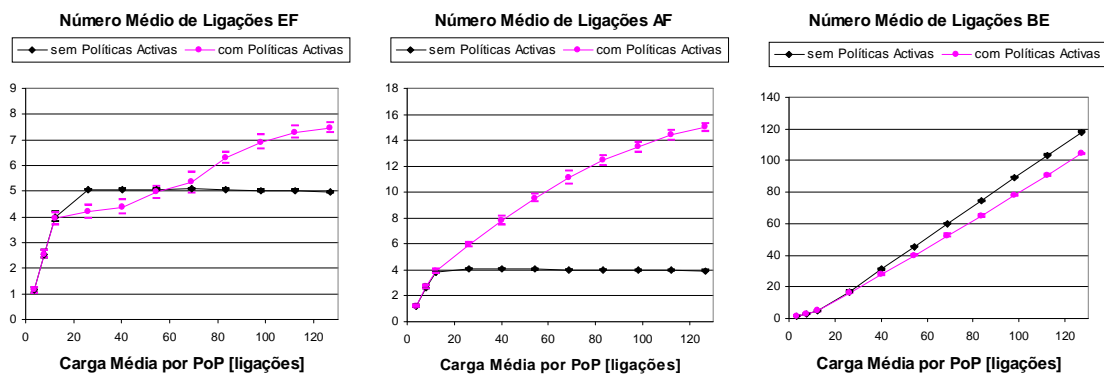


Figura 5.44: Comparação do número de ligações nas situações com e sem as políticas activas de gestão.

A figura 5.45 ilustra a fracção das ligações com a classe de tráfego degradada com e sem a utilização conjunta das políticas activas. Da observação destas figuras, conclui-se que a disponibilidade do serviço proporcionado pela classe EF é sempre pelo menos 98%, que é muito elevada. A disponibilidade do serviço proporcionado pela classe AF é sempre pelo menos 85%, que é relativamente elevada. Cumprem-se assim os requisitos relativos à disponibilidade dos serviços. As melhorias obtidas são principalmente devidas à política activa de gestão do mapeamento para PHBs, à custa de apenas permitir que as aplicações mais importantes usem as classes de tráfego mais prioritárias.

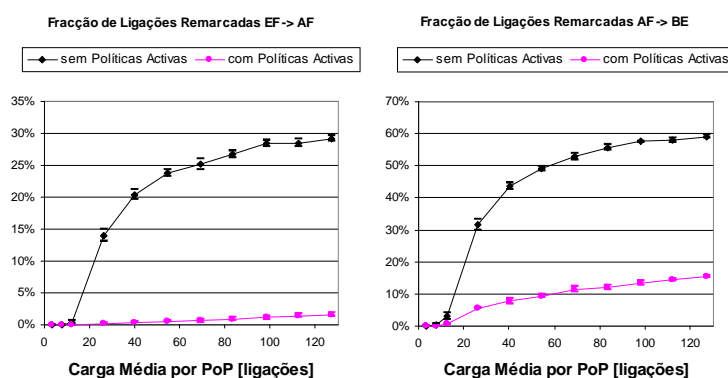


Figura 5.45: Comparação da fracção do número de ligações com a classe de tráfego degradada nas situações com e sem as políticas activas de gestão.

A figura 5.46 ilustra o débito médio (em bit/segundo) obtido com e sem a utilização conjunta das políticas activas. A utilização do corretor de largura de banda em

EF e a inexistência de descarte de pacotes dentro de perfil significam que o débito especificado no contrato é assegurado.

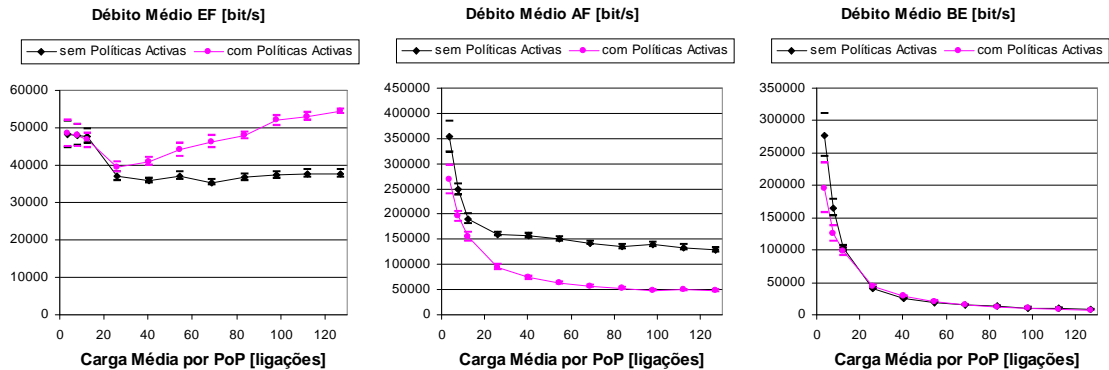


Figura 5.46: Comparação do débito nas situações com e sem as políticas activas de gestão.

A figura 5.47 ilustra o *jitter* médio (em segundos) obtido com e sem a utilização conjunta das políticas activas. O *jitter* obtido com a utilização de políticas activas é em média superior, principalmente devido à redução da largura de banda, mas na classe EF, para cargas altas, é inferior, devido à predominância de aplicações CBR nesta situação.

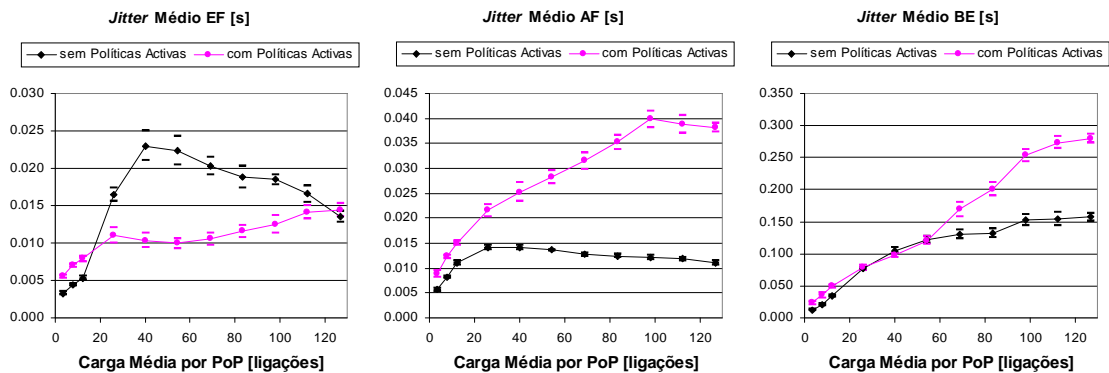


Figura 5.47: Comparação do *jitter* nas situações com e sem as políticas activas de gestão.

Para avaliar a eficácia no cumprimento dos SLAs, repetiram-se as simulações para diferentes valores dos SLAs. A tabela 5.13 ilustra os valores obtidos para os máximos do atraso nas classes EF e AF, fazendo variar a carga oferecida à rede. Como medida do custo para o fornecedor de serviço, apresenta-se a largura de banda média no

núcleo da rede, média esta calculada com os valores obtidos para as diferentes cargas oferecidas à rede. Decidiu-se ordenar a tabela por ordem crescente deste parâmetro.

| <i>SLA(atrasoEF)</i> | <i>SLA(atrasoAF)</i> | <b>Máximo atrasoEF</b> | <b>Máximo atrasoAF</b> | <b>Largura de Banda Média</b> |
|----------------------|----------------------|------------------------|------------------------|-------------------------------|
| 60 ms                | 200 ms               | 57.8 ms                | 140.4 ms               | 2.723 Mbps                    |
| 60 ms                | 120 ms               | 56.4 ms                | 104.9 ms               | 2.921 Mbps                    |
| 100 ms               | 100 ms               | 65.0 ms                | 88.9 ms                | 2.972 Mbps                    |
| 100 ms               | 200 ms               | 76.6 ms                | 168.6 ms               | 3.066 Mbps                    |
| 200 ms               | 800 ms               | 75.4 ms                | 209.0 ms               | 3.186 Mbps                    |
| 50 ms                | 100 ms               | 50.5 ms                | 85.4 ms                | 3.476 Mbps                    |
| 45 ms                | 90 ms                | 48.4 ms                | 77.8 ms                | 4.476 Mbps                    |
| 40 ms                | 70 ms                | 45.5 ms                | 64.4 ms                | 6.738 Mbps                    |
| 35 ms                | 50 ms                | 40.2 ms                | 50.9 ms                | 16.840 Mbps                   |
| 25 ms                | 30 ms                | 38.9 ms                | 39.6 ms                | 35.537 Mbps                   |

Tabela 5.13: Resultados obtidos com diferentes valores do SLA para o atraso.

Dos resultados obtidos, conclui-se que é possível ajustar os valores dos SLAs, dentro de certos limites. O caso com 25 ms para o *SLA(atrasoEF)* e 30 ms para o *SLA(atrasoAF)* representa o caso limite demasiado bom, que não é possível atingir, devido à existência de tempos de propagação que não se podem reduzir mais. O caso com 200 ms para o *SLA(atrasoEF)* e 800 ms para o *SLA(atrasoAF)* representa o caso limite demasiado mau, que também não é atingido. Na realidade, o atraso máximo em EF não cresce muito, visto que o corretor de largura de banda impõe um limite ao número máximo de ligações que podem usar esta classe. Já no caso de AF, são possíveis variações muito maiores, pois utilizou-se uma política activa de gestão de admissão de ligações.

Da análise da tabela, pode-se dizer que é possível ajustar o *SLA(atrasoEF)* entre 40 ms e 60 ms, obtendo-se um erro inferior a 8%, e ajustar o *SLA(atrasoAF)* entre 50 ms e 120 ms, com um erro inferior a 15%. Na realidade, da observação da figura 5.42, verifica-se que em alguns casos o erro é maior, consoante a carga da rede. Estas diferenças sugerem que os limiares de decisão utilizados poderiam ser melhor afinados. Voltar-se-á a este tópico na subsecção seguinte (5.5.2).

Verifica-se ainda, como seria natural, haver uma certa dependência entre os atrasos obtidos em EF e em AF. Na realidade, mantendo um dos SLAs fixo, e variando

o outro, resulta numa variação no atraso obtido, mesmo para aquele em que não se modificou o SLA. No entanto, esta dependência é muito menor num sentido: uma mudança no  $SLA(atrasoAF)$  resulta numa menor variação no  $SLA(atrasoEF)$  do que no sentido contrário.

Da análise desta tabela, verifica-se que, se for possível fazer cumprir os SLAs, a largura de banda média no núcleo é inferior à situação sem políticas activas, libertando recursos para outras utilizações.

Paradoxalmente, verifica-se que relaxar os limites impostos nos SLAs resulta num ligeiro aumento da largura de banda. Tal é explicado pelo facto de ser possível admitir um maior número de ligações em AF, devido à política activa de gestão de admissão de ligações. Assim, para manter a taxa de descarte de pacotes AF muito baixa, torna-se necessário aumentar a largura de banda média. Mas, nestes casos, os custos para o operador da rede são compensados por um aumento nos lucros, devido ao maior número de ligações aceites.

Por outro lado, se os SLAs impostos não forem possíveis de fazer cumprir, ou estiverem perto disso, então a largura de banda tende rapidamente para o máximo possível, resultando em elevados custos para o operador.

Finalmente, conclui-se que a utilização das políticas activas é uma excelente forma de conseguir fazer cumprir os SLAs, minimizando os custos de operação da rede.

### **5.5.2 Política Activa de Planeamento**

A política activa de planeamento tem como função avaliar o cumprimento dos SLAs numa escala de tempo maior, e ajustar os parâmetros da política activa de gestão de SLAs, de forma a que os SLAs sejam cumpridos com maior eficiência.

O principal problema notado nos resultados apresentados na subsecção anterior é a existência de um desvio entre os valores dos atrasos obtidos e os valores pretendidos nos SLAs. Caso se afinem os limiares de actuação das várias políticas, deve ser possível obter melhores resultados. Uma forma mais fácil de afinar estes limiares de actuação é ajustar os próprios valores dos SLAs que são utilizados nas políticas activas, dos quais os limiares de actuação dependem.

O apêndice B apresenta uma introdução aos controladores PID. Desta introdução, conclui-se que a utilização de uma malha de realimentação com um

integrador, para ajustar os valores dos SLAs utilizados pelas políticas activas, deve contribuir para a eliminação do erro estático entre os atrasos pretendidos e os obtidos.

As teorias apresentadas no apêndice B são adequadas para sistemas lineares e invariantes no tempo. No entanto, o sistema composto pela rede e o conjunto de políticas activas associadas não é linear, nem invariante no tempo. Além disso, há uma dependência entre os resultados obtidos para as diferentes classes de tráfego. Contudo, o sistema exhibe alguma linearidade dentro da gama de valores em que é possível fazer cumprir os SLAs. Assim, utilizaram-se dois controladores, utilizando a regra `pid_control`, para afinar os valores do  $SLA(atrasoEF)$  e  $SLA(atrasoAF)$ , conforme ilustrado na figura 5.48.

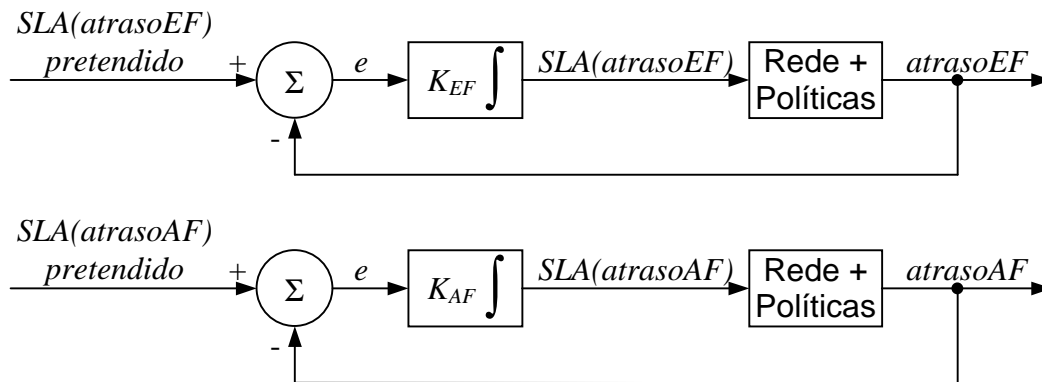


Figura 5.48: Sistema de controlo dos SLAs.

Não é objectivo do controlador aumentar a rapidez de convergência do sistema, até por que isso teria custos em termos de largura de banda necessária, e ainda efeitos negativos em variações nos atrasos, aumentando portanto o *jitter*. Desta forma, apenas se utilizaram controladores integrativos, sendo a equação às diferenças respectiva, válida tanto para EF, como para AF, dada por:

$$SLA(n) = SLA(n-1) + K.[SLA_{pretendido} - atraso(n)] \quad (5.15)$$

onde  $n$  representa o instante de amostragem actual.

Como nem sempre é possível eliminar totalmente o erro existente, o integrador pode ter tendência a fazer aumentar, ou diminuir, indefinidamente o valor do SLA. Este comportamento fornece uma indicação de que não é possível atingir os objectivos pretendidos para os SLAs, o que é útil para a revisão dos SLAs. Embora o aumento ou a diminuição excessiva dos SLAs aplicados à rede não prejudique o funcionamento das

políticas activas, optou-se por introduzir um limitador à saída do integrador, por forma a manter o valor dos SLAs dentro de uma determinada gama. Configurou-se o limitador para limitar cada SLA entre 0.25 e 4 vezes o valor pretendido para o SLA correspondente.

Para configurar cada controlador integrativo é necessário fixar dois parâmetros: a frequência de amostragem e o ganho do integrador.

Como os SLAs são normalmente verificados ao fim de um prazo longo, tipicamente um mês, o período de amostragem necessário deveria ser de algumas horas. No que diz respeito ao ganho do integrador, pretende-se que ele seja baixo, para que a afinação dos SLAs seja lenta, e assim evitar os efeitos negativos no custo e no *jitter*. Quando se aumenta o ganho do integrador, há tendência para fazer correcções nos SLAs grandes de mais, podendo originar oscilações. Quando se diminui o ganho do integrador, as correcções são mais pequenas, pelo que demora mais tempo a corrigir o erro no cumprimento do SLA, o que também é uma desvantagem, embora menos grave.

Para não tornar a simulação muito demorada, e atendendo a que as políticas activas actuam a cada segundo, decidiu-se utilizar um período de amostragem de 5 segundos para os controladores. Para o ganho do integrador, deve-se ter um ganho várias vezes menor do que 1, de maneira a que o erro estático demore vários períodos de amostragem a ser compensado, garantindo assim as variações lentas na afinação dos SLAs. Utilizou-se um ganho do integrador de 0.25 para ambos os controladores, tendo-se verificado experimentalmente ser um valor adequado, pois permitia uma convergência rápida do sistema, praticamente sem oscilações.

Não se observou vantagem em incluir termos proporcionais e derivativos nos controladores. Na realidade, os valores dos atrasos na rede têm variações muito rápidas comparativamente com o período de amostragem escolhido, pelo que incluindo termos proporcionais e derivativos nos controladores, ficar-se-ia com controladores mais rápidos, que responderiam às perturbações de alta frequência no atraso, o que é indesejável. Utilizando apenas o termo integrativo, os controladores ficam mais lentos, respondendo apenas ao valor médio. Adicionalmente, para reduzir mais as variações, não se tira apenas uma amostra do valor do atraso, mas faz-se uma média durante o período de amostragem.

A figura 5.49 compara o atraso médio (em segundos) obtido nas várias classes de tráfego utilizando a política activa de planeamento com a situação em que não se

utiliza esta política, mantendo-se apenas o funcionamento das outras políticas activas (correspondendo aos resultados da subsecção anterior (5.5.1)). Para a elaboração destes gráficos, foram feitas simulações mais longas, de 180 segundos, de forma a dar tempo a que a política activa de planeamento actue efectivamente.

Da observação desta figura, conclui-se que a utilização desta política de planeamento melhora os resultados obtidos, aproximando-os dos SLAs pretendidos, principalmente na classe AF, onde há maior flexibilidade no ajuste do atraso.

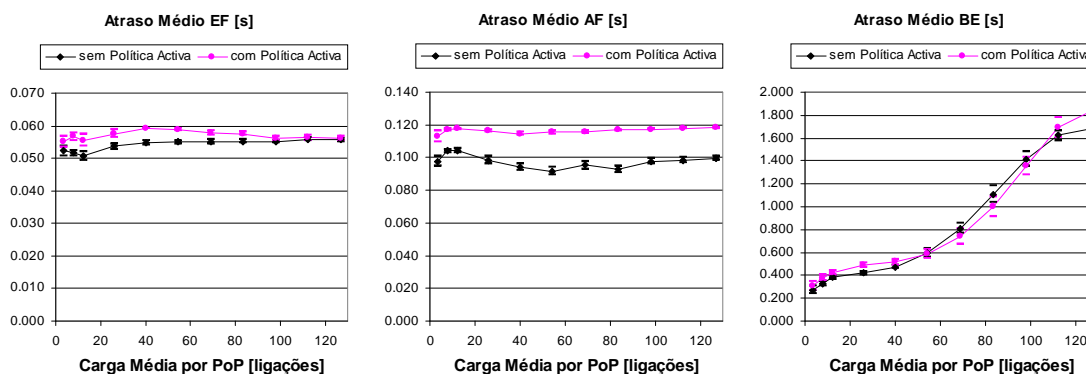


Figura 5.49: Comparação do atraso nas situações com e sem a política activa de planeamento.

Repetiram-se as simulações apresentadas na tabela 5.13, agora com a utilização da política activa de planeamento, apresentando-se os resultados respectivos, pela mesma ordem, na tabela 5.14.

| <i>SLA(atrasoEF)</i> | <i>SLA(atrasoAF)</i> | Máximo <i>atrasoEF</i> | Máximo <i>atrasoAF</i> | Largura de Banda Média |
|----------------------|----------------------|------------------------|------------------------|------------------------|
| 60 ms                | 200 ms               | 59.5 ms                | 188.3 ms               | 3.081 Mbps             |
| 60 ms                | 120 ms               | 59.5 ms                | 118.3 ms               | 2.902 Mbps             |
| 100 ms               | 100 ms               | 69.6 ms                | 98.4 ms                | 2.960 Mbps             |
| 100 ms               | 200 ms               | 75.7 ms                | 195.6 ms               | 3.299 Mbps             |
| 200 ms               | 800 ms               | 75.6 ms                | 210.3 ms               | 3.423 Mbps             |
| 50 ms                | 100 ms               | 49.8 ms                | 99.5 ms                | 3.197 Mbps             |
| 45 ms                | 90 ms                | 45.2 ms                | 88.5 ms                | 4.470 Mbps             |
| 40 ms                | 70 ms                | 40.5 ms                | 69.4 ms                | 8.843 Mbps             |
| 35 ms                | 50 ms                | 36.6 ms                | 49.9 ms                | 28.210 Mbps            |
| 25 ms                | 30 ms                | 35.8 ms                | 36.4 ms                | 38.988 Mbps            |

Tabela 5.14: Resultados obtidos com diferentes valores do SLA para o atraso e utilização da política activa de planeamento.



Da análise da tabela, pode-se dizer que é possível agora ajustar o  $SLA(atrasoEF)$  entre 40 ms e 60 ms, obtendo-se um erro inferior a 1.3%, e ajustar o  $SLA(atrasoAF)$  entre 50 ms e 120 ms, com um erro inferior a 1.7%.

Da comparação desta tabela com a anterior, podem-se tirar algumas conclusões.

Os resultados obtidos com a política activa de planeamento são cerca de uma ordem de grandeza mais precisos do que os obtidos sem esta política, demonstrando a utilidade da política activa de planeamento.

Nos casos em que o atraso obtido baixou, foi quase sempre necessário utilizar mais largura de banda, como seria natural. No entanto, verifica-se novamente que, relaxando os limites impostos nos SLAs, se obtém um aumento da largura de banda. Assim acontece, pois a política activa de gestão de admissão de utilizadores vai deixar entrar mais utilizadores na classe AF. Para não causar descarte de pacotes dentro de perfil nesta classe, a largura de banda tem de aumentar. Porém, do ponto de vista do operador da rede, este aumento da largura de banda não resulta em prejuízo, visto que, havendo mais utilizadores, vai-se cobrar mais.

O funcionamento da política activa de planeamento pode ser interpretado como estando a ajustar continuamente o ponto de funcionamento médio para as classes de tráfego EF e AF na curva da figura 5.50, que representa o atraso médio sofrido pelos pacotes em cada fila de espera (igual à anterior figura 5.8). Para a classe BE, não é feito nenhum esforço de fixar o ponto de funcionamento médio, pelo que se obtém o que for possível com os recursos que sobram das outras classes, o que corresponde de facto a um serviço melhor esforço.

Conclui-se que, no caso dos atrasos, é possível conseguir obter a QoS pretendida dentro de certos limites, com uma precisão razoável, usando a política activa de planeamento. Nos casos em que se impõe um limite para o atraso impossível de fazer cumprir, as políticas activas actuam no sentido de aproximar os resultados o mais possível dos limites impostos, não havendo problemas de funcionamento.

Seria possível desenvolver outras políticas de planeamento que agissem sobre outros parâmetros, como por exemplo, permitindo ao gestor da rede controlar de forma automática uma maior ou menor admissão de ligações, a utilização de linhas extra, ou a largura de banda das linhas, consoante os custos que se admitisse ter, e a QoS que se pretendesse oferecer aos utilizadores.

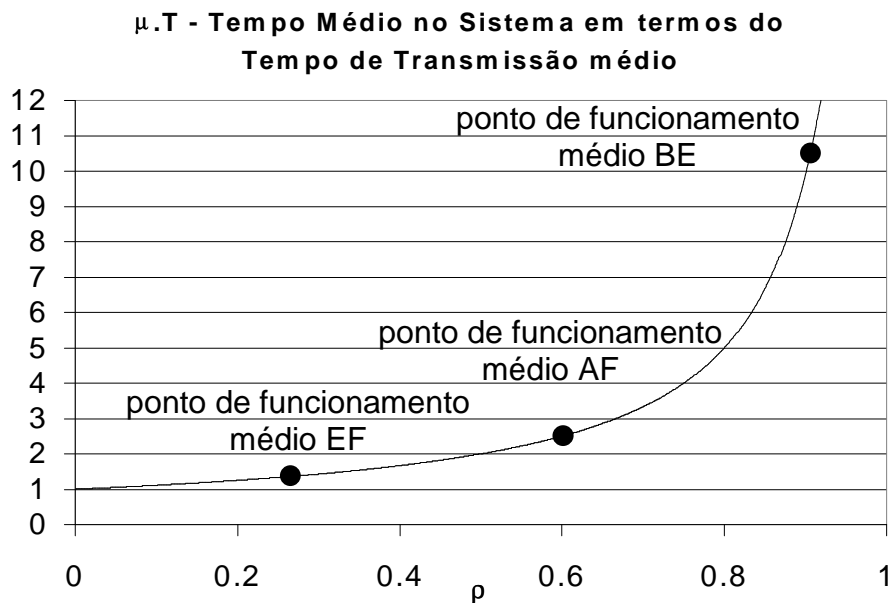


Figura 5.50: Ponto de funcionamento médio para as diferentes classes de tráfego.

## 5.6 AVALIAÇÃO

Na secção 3.5.4 apresentaram-se alguns critérios com o objectivo de avaliar a eficiência, justiça, distribuição e convergência de um sistema de gestão. Nesta secção avalia-se o cenário estudado de acordo com cada um deles em maior pormenor, deixando-se para o próximo capítulo uma avaliação mais geral do modelo de gestão sem entrar em detalhes de cenários particulares.

A eficiência traduz a aproximação ao valor ideal. Verificou-se que foi possível fazer cumprir os requisitos especificados nos contratos dos vários níveis de abstracção. Os SLAs de atraso são cumpridos com um erro relativamente pequeno (da ordem de 1%), desde que os valores pretendidos sejam atingíveis. O débito pretendido em cada ligação é assegurado. Não há descarte de pacotes na classe EF, e o descarte é muito raro para pacotes AF dentro de perfil. Na realidade, só se verificou haver algum descarte de pacotes AF dentro de perfil para algumas simulações de carga mais elevada (taxa de descarte de pacotes inferior a  $10^{-4}$ , para cargas superiores a 100 ligações por PoP). A disponibilidade do serviço é muito grande, sendo cerca de 100% para situações sem sobrecarga da rede, e mantendo-se em cerca de 97% para EF e 84% para AF, com uma

sobrecarga de 8 vezes. Por último, a QoS é melhor em EF do que em AF, e melhor em AF do que em BE, tal como se pretendia.

A justiça mede o grau com que os recursos são equitativamente repartidos pelos utilizadores da mesma classe. Deste ponto de vista, não interessa apenas a classe de tráfego utilizada por cada utilizador, mas também o instante e a localização onde ele e o seu destinatário se encontram, visto que todos os parâmetros de QoS podem depender significativamente do tempo e da localização. Na realidade, dois utilizadores que estejam no mesmo nó e usem a mesma classe de tráfego podem obter débitos muito diferentes, caso um tenha uma ligação local, e o outro uma de longa distância, ou um utilize uma hora de vazio, e o outro uma hora de ponta. No entanto, se ambos tiverem simultaneamente o mesmo destino, são tratados de forma absolutamente idêntica pela rede, pelo que devem obter uma QoS muito semelhante.

Do ponto de vista da distribuição, pretende-se minimizar a informação de gestão transmitida na rede. Esta foi uma preocupação constante ao longo da concepção do sistema de gestão. De facto, ao colocar políticas activas que podem actuar com uma certa independência junto dos equipamentos, permite-se eliminar parte do tráfego de gestão. Assim, as situações em que estas políticas necessitam de trocar informação com políticas globais de nível superior são: transferência de informação para calcular os parâmetros de QoS globais à rede, envolvidos nos SLAs, cuja periodicidade num sistema real não é habitualmente muito elevada, como se viu anteriormente na secção 2.7; situações de excepção, em que é necessário enviar alarmes para cima; e situações em que as políticas de nível superior necessitam de ajustar algum parâmetro nas políticas de nível inferior, o que é relativamente raro. Na realidade, as políticas activas que trabalham com informação global têm uma perspectiva mais estratégica, pelo que a informação de gestão necessária é reduzida e de nível semântico elevado. De facto, esta informação é tratada localmente, nas políticas activas colocadas junto dos equipamentos, antes de ser enviada para as políticas de nível superior.

A convergência mede a rapidez e a suavidade com que se chega ao resultado pretendido. Do ponto de vista de rapidez, procurou-se ter uma resposta rápida a situações de falha, ou de falha iminente, e uma resposta lenta, para minimizar as perturbações, no sentido de regresso à normalidade. Ou, em termos de QoS, procurou-se melhorar rapidamente a QoS sempre que é detectada a sua degradação, e reduzi-la lentamente quando é melhor do que o estritamente necessário, de acordo com o

algoritmo genérico de ajuste de parâmetros do sistema descrito anteriormente na figura 3.10. Pelo facto de as políticas activas actuarem perto dos objectos geridos, minimiza-se o seu tempo de reacção, e reduz-se ao mínimo a possibilidade de estar a actuar em sentido contrário ao devido, maximizando-se assim as garantias de convergência. No entanto, como o tráfego numa rede é caracterizado por frequentes rajadas, será natural esperar que, com a utilização de políticas activas, as perturbações sentidas pelo tráfego de uma dada fonte (por exemplo em termos de *jitter*) sejam maiores do que no caso em que a rede é dimensionada pelo pico de tráfego e nunca sofre alterações.

## 5.7 CONCLUSÕES

O exemplo descrito neste capítulo mostra como se pode avaliar a qualidade de serviço que os utilizadores estão a obter. Mostra-se ainda, como controlando um conjunto de parâmetros chave em diferentes níveis de abstracção, as políticas activas podem automatizar tarefas de gestão, melhorando o desempenho do sistema, de forma a cumprir os SLAs impostos. Mostra-se também, como utilizando planeamento, com base num controlador integrativo, se podem ajustar os limiares de decisão das políticas activas, de forma a fazer cumprir os SLAs com mais exactidão.

A metodologia proposta para a construção de sistemas de gestão por políticas activas permite atacar os problemas de gestão de uma forma sistemática. As políticas activas possibilitam a automatização de tarefas de gestão em tempo real, auxiliando o cumprimento dos objectivos de gestão e libertando o gestor da rede para tarefas de mais alto nível.

É importante realçar que o comportamento das políticas activas foi construído utilizando as regras de actuação pré-definidas, mostrando a sua reutilização nas mais diversas situações, por simples mudança da parametrização. A excepção foi a política activa de gestão de linhas extra, onde foi programado um algoritmo específico.

# CAPÍTULO 6

## AVALIAÇÃO

Neste capítulo avalia-se o modelo e os cenários propostos, comparando-os com as alternativas existentes na literatura.

### 6.1 COMPARAÇÃO DO MODELO

Na literatura têm sido propostos numerosos modelos e arquitecturas de gestão. Na maior parte dos casos, as arquitecturas limitam-se a fazer monitorização. Um exemplo é [Anerousis 99], que utiliza vários níveis hierárquicos para obter escalabilidade na recolha de informação de MIBs. A informação recolhida é utilizada em expressões que dão uma visão da rede a um nível de abstracção superior. No entanto, nada é dito sobre como relacionar esta informação com QoS e SLAs, ou como usar esta informação para gerir a rede.

Outros autores vão mais além, ao utilizar a informação recolhida para detecção de falhas (e.g. [Ho 99], [Franceschi 96]) e geração de eventos (e.g. [Al-Shaer 99]).

Um passo adicional é utilizar a informação recolhida para avaliar o estado da rede, ou a QoS que está a ser obtida (e.g. [Goldszmidt 93], [Cherkaoui 98]). Este passo pode ser combinado com detecção de falhas ou com funcionalidades semelhantes às disponibilizadas pelas SMFs: registo de eventos, filtragem de eventos, alarmes, ou estatísticas (e.g. [Stamatelopoulos 95], [Stamatelopoulos 97], [Lewis 97], [Darst 99]). A vantagem de combinar detecção de falhas com QoS é permitir detectar situações anormais antes de as falhas causarem problemas graves [Thottan 98].

Alguns trabalhos abrangem a avaliação de SLAs. Por exemplo, [Bissel 00] aplica conceitos de SLM para avaliar serviços de correio electrónico e WWW, mas limita-se a medir os níveis de serviço obtidos. Em [Asawa 98] é proposta uma metodologia de monitorização de níveis de serviço com base na avaliação passiva do débito em *proxies* WWW, visto que a maior parte dos serviços são baseados em WWW.

Em [Bhoj 01] é descrita uma forma de definir contratos para SLAs, e uma arquitectura para os verificar, mas nada é dito sobre como agir sobre a rede para assegurar os contratos.

Conceitos de fluxos de trabalho (*workflow*) também podem ser utilizados para modelar processos de negócio relativos a SLAs [Schmidt 00]. No entanto, a metodologia de fluxos de trabalho abrange conceitos como escalonamento e coordenação de tarefas, sendo mais adequada para processos de negócio que não sejam de tempo real. Assim, no exemplo dado nesse trabalho, o cliente, ao detectar uma falha, notifica o fornecedor para este desencadear as tarefas necessárias à sua resolução, não se explicando como assegurar SLAs. Os conceitos de fluxos de trabalho também podem ser combinados com QoS [Cardoso 02]. Mas este trabalho limita-se a propor métricas para avaliar a QoS obtida pelos utilizadores, com base na QoS das tarefas que constituem os fluxos de trabalho, reconhecendo a escassez de trabalhos na área de QoS para sistemas de gestão de fluxos de trabalho.

Em [Alexandrov 97] é proposto um método de construção de modelos de QoS de aplicações distribuídas, aplicado a serviços WWW, semelhante à metodologia proposta nesta tese para decomposição dos processos de negócio. O método proposto em [Alexandrov 97] tem vários passos. É construído um modelo das tarefas com a decomposição do processamento do serviço em sequências de tarefas. É construído um modelo dos recursos com o conjunto dos recursos necessários à execução do serviço. É construído um modelo de mapeamento de tarefas para recursos. Para cada conjunto de tarefa e recurso associado, é construído um modelo de desempenho de filas de espera. Finalmente, tenta-se associar parâmetros de desempenho do serviço a parâmetros controláveis do modelo. Comparando com o proposto nesta tese, notam-se algumas diferenças, pois nesta tese não se utiliza necessariamente um modelo de filas de espera, pretendendo-se antes identificar parâmetros dos objectos geridos que possam ser associados à QoS e parâmetros que possam ser modificados para obter a QoS pretendida.

Mais próximo do trabalho desta tese estão os trabalhos que envolvem a utilização de políticas de gestão. Os trabalhos de [Wies 94a], [Sloman 94], [Boutaba 95] e [Koch 96] apresentam hierarquias de políticas. Contudo, estes trabalhos não se integram com gestão de níveis de serviço, nem refinam os requisitos por diferentes níveis de abstracção de políticas actuando sobre a rede até aos equipamentos, nem

permitem uma fácil reutilização de políticas, ao contrário do modelo proposto nesta tese.

A maioria das linguagens para construir aplicações de gestão são simples interfaces de programação para o protocolo de gestão. As regras de actuação propostas nesta tese, por outro lado, estão orientadas para controlar e ajustar os parâmetros de QoS, em qualquer nível de abstracção, de forma a atingir os objectivos de gestão pretendidos. Como se viu, estas regras de actuação são facilmente reutilizáveis em diferentes situações e diferentes níveis de abstracção, bastando modificar os parâmetros dos objectos geridos que são lidos e modificados, bem como os pesos com que cada parâmetro contribui para cada modificação a realizar. As políticas activas construídas com base nestas regras de actuação permitem automatizar as tarefas de gestão, sendo adequadas para resolver uma gama larga de problemas de gestão de desempenho, gestão de níveis de serviço, gestão de falhas e planeamento.

## 6.2 COMPARAÇÃO DOS CENÁRIOS

Na literatura também têm sido propostos diversos tipos de cenários onde se pretende assegurar alguma QoS aos utilizadores. Uma vez que a introdução de conceitos de QoS só por si não cria recursos, há que utilizar mecanismos para repartir os recursos pelos utilizadores de acordo com o nível de QoS pretendido para cada um. Assim, é normal haver cenários onde se utilizam tecnologias de rede tais como ATM, IEEE 802.1p, IntServ, ou DiffServ, como forma de dividir os recursos da rede entre os utilizadores.

[Puka 99] apresenta um exemplo de SLAs para redes ATM, mas limita-se a monitorizar e avaliar os parâmetros de QoS específicos da tecnologia ATM e a disponibilidade da rede.

[Y.1541] propõe-se normalizar objectivos de desempenho para redes IP, definindo 6 classes de QoS para utilização em acordos de serviço. São definidos limites extremo a extremo para o atraso (100 ms, 400 ms, 1 s, ou sem limite), variação do atraso (50 ms, ou sem limite), perdas ( $10^{-3}$ , ou sem limite), e pacotes errados ( $10^{-4}$ , ou sem limite). É ainda proposta uma correspondência destas classes de QoS para classes de tráfego DiffServ. No entanto, nada é indicado sobre como atingir os objectivos

propostos, sendo referido que os valores propostos são provisórios, estando sujeitos a revisão consoante a experiência operacional obtida.

[Gibbens 00] apresenta um exemplo de SLAs para serviços diferenciados, mas estuda o tráfego numa rede com apenas uma linha, limitando-se a analisar em que condições é que os requisitos de qualidade podem ser assegurados.

Em [Ramanathan 01] é proposto um mecanismo activo de gestão de recursos num ambiente de serviços diferenciados que actua dinamicamente, com base no estado actual da rede e nas necessidades das aplicações. De acordo com este mecanismo, quando o corretor de largura de banda detecta que os clientes subutilizam a largura de banda que reservaram, coloca-os numa classe inferior, libertando parte dos recursos reservados. As simulações realizadas mostram que este mecanismo activo poupa entre 50% e 75% da largura de banda, oferecendo aos clientes a QoS requerida. Este exemplo está próximo do funcionamento das políticas activas de gestão de admissão de ligações do cenário do capítulo 5. Assim, representa uma solução particular num dado nível de abstracção, não se enquadrando numa arquitectura e metodologia globais para resolver problemas de SLM. Pelo facto de ser um algoritmo isolado, se a QoS não for assegurada por algum motivo, o SLA falha imediatamente, não havendo nenhum mecanismo de recurso noutra nível de abstracção que possa então agir, ao contrário do caso desta tese.

O projecto Tequila (*Traffic Engineering for Quality of Service in the Internet, at Large Scale*) [Tequila] é um dos vários projectos recentes na área de engenharia de tráfego. O projecto Tequila tem como objectivo estudar, especificar, implementar e validar um conjunto de definições de serviço e ferramentas de engenharia de tráfego para obter garantias de QoS quantitativas extremo a extremo, através de um cuidadoso dimensionamento, controlo de admissão e gestão dinâmica de recursos, em redes com serviços diferenciados. [Trimintzios 01] apresenta um cenário de gestão de qualidade de serviço em redes com serviços diferenciados. São utilizados mecanismos de engenharia de tráfego com o objectivo de otimizar a configuração da rede e o seu desempenho. Estes incluem gestão dinâmica de recursos, tais como a configuração do espaço nas filas de espera e parâmetros dos mecanismos de serviço dos *routers*; e gestão dinâmica de rotas, através de MPLS (*Multiprotocol Label Switching*), ou da restrição de certas classes de tráfego em certas interfaces. Este exemplo está próximo do funcionamento de algumas políticas activas do capítulo 5. Nota-se a existência de vários mecanismos complementares para assegurar a qualidade de serviço, que não existiam no exemplo



precedente. No entanto, parece ser necessário manter informação sobre os caminhos seguidos por todos os fluxos de dados na rede, o que reduz a escalabilidade e contraria a perspectiva de tráfego agregado de DiffServ. Além disso, não há um mecanismo de planeamento automático para ajustar dinamicamente os parâmetros dos outros mecanismos de gestão, ao contrário do proposto nesta tese, de que é exemplo a política activa de planeamento da secção 5.5.2.

Outro trabalho relevante é a iniciativa QBone do projecto Internet2 [QBone] [Teitelbaum 99]. Embora inicialmente este projecto tenha procurado criar um serviço *Premium* baseado na classe EF de DiffServ, dificuldades de implementação levaram a que este serviço tenha sido congelado. Na realidade, um serviço deste tipo exige uma actualização de todos os equipamentos, modificações dramáticas tanto na operação da rede, como nas interacções entre entidades pares e nos modelos de negócio, faltando ainda formas de verificar o funcionamento do serviço tanto por parte dos utilizadores como da rede. Assim, este projecto tem recentemente focado no desenvolvimento de serviços não elevados, tais como o QBSS (*QBone Scavenger Service*), que permite que os utilizadores marquem tráfego a ter um serviço degradado em caso de congestão, e o ABE (*Alternative Best Effort*) [Hurley 01], que permite aos utilizadores escolher entre baixo atraso com ritmo limitado e um maior débito com maior atraso. Este novo foco está mais próximo do trabalho realizado nesta tese, em que as políticas activas actuam no sentido de fazer cumprir os limites de QoS impostos, mas sem oferecer garantias absolutas de que o limites serão cumpridos.

## 6.3 CONCLUSÕES

Nesta tese integraram-se os conceitos de SLM, QoS e políticas de gestão apresentados no capítulo 2. Para além de se avaliar a QoS obtida, em diferentes níveis de abstracção, actua-se sobre os sistemas geridos no sentido de melhorar o nível de serviço obtido até ao especificado num SLA. Pelo facto de haver vários níveis de políticas activas, pode-se trabalhar transparentemente no nível de abstracção adequado, podendo problemas nos níveis mais baixos ser resolvidos automaticamente pelos níveis mais altos, sem intervenção humana.



# CAPÍTULO 7

## CONCLUSÃO

Neste capítulo sintetiza-se o trabalho desenvolvido avaliando-se as respectivas contribuições, apresentam-se as conclusões, e apontam-se sugestões de extensões e indicações para investigação futura.

### 7.1 SÍNTESE

Nesta tese foi apresentado um modelo de gestão, baseado em políticas activas, para gerir redes e sistemas, com o objectivo de fazer cumprir objectivos de gestão e requisitos de qualidade de serviço em diferentes níveis de abstracção. Foi proposta uma metodologia para analisar os requisitos de qualidade de serviço dos processos de negócio de empresas e mapeá-los nos diferentes níveis de abstracção.

Este modelo foi aplicado a dois cenários, um de uma rede empresarial, no capítulo 4, e outro de um fornecedor de serviços Internet, no capítulo 5. Demonstrou-se, nestes exemplos, como o modelo proposto pode ser aplicado para analisar problemas de gestão, e como se podem utilizar políticas activas para assegurar requisitos de qualidade de serviço.

Para construir o comportamento das políticas activas, foi proposto um conjunto de regras de actuação (secção 3.6), que se provou serem úteis na construção da maior parte das políticas activas utilizadas. Estas regras podem ser reutilizadas para diferentes situações, bastando apenas mudar os parâmetros dos sistemas que as regras controlam e a própria configuração das regras, para as adaptar a diferentes situações. A excepção foi a política activa de gestão de linhas extra (secção 5.4.4), que exigiu programação específica para actuar no caso de falhas de linhas.

A dificuldade prática de definir os limiares de actuação das regras das políticas activas, de forma a fazer cumprir os objectivos de gestão com exactidão, pode ser ultrapassada utilizando uma política activa de planeamento com um controlador

integrativo, como foi demonstrado na secção 5.5.2. A detecção da saturação do integrador é importante, pois indica que os objectivos pretendidos não são realizáveis.

## 7.2 CONCLUSÕES

Dos resultados apresentados, podem-se tirar algumas conclusões:

- As políticas activas são úteis para otimizar o desempenho das redes e sistemas geridos, ajustando automaticamente os seus parâmetros, de forma a atingir os objectivos de gestão.
- As políticas activas permitem avaliar a qualidade de serviço que está a ser obtida pelos utilizadores, e assegurá-la dentro dos limites do possível, podendo ser utilizadas para fazer cumprir SLAs e objectivos de gestão.
- As políticas activas podem, na maior parte dos casos, ser construídas com base no conjunto de regras de actuação proposto.
- A hierarquia de políticas activas proporciona vários graus de abstracção, com grande flexibilidade, autonomia, melhorada escalabilidade e tolerância a falhas, relativamente a alternativas não hierárquicas.
- O poder de actuação das políticas activas passa por ter vários parâmetros nos sistemas geridos que são manuseados pelos níveis de abstracção mais baixos. A complexidade introduzida, pela existência de um número elevado de parâmetros, acaba por ser eliminada pelo facto de as políticas activas os gerirem, dando uma visão mais abstracta para os níveis superiores. O cenário do capítulo 5 é a prova da viabilidade do funcionamento de uma arquitectura deste género.
- A metodologia proposta para a construção de sistemas de gestão por políticas activas permite atacar os problemas de gestão de uma forma sistemática. No entanto, esta metodologia exige trabalho especializado de quem conheça muito bem os sistemas envolvidos, o que não é fácil. Actualmente, está-se muito longe de apenas integrar blocos já feitos e pôr o sistema de gestão a funcionar de imediato. Mas, depois de construído o sistema de gestão por políticas activas, pode-se ganhar muito com a automatização da gestão.
- A maior parte das garantias descritas são do tipo melhor esforço, pois para ter garantias determinísticas, ou estatísticas, é necessário ter um bom modelo do sistema

gerido, que normalmente não se tem. Por esta razão, a realizabilidade das garantias é difícil de avaliar sem recorrer ao sistema real, ou a uma simulação do sistema real.

- No caso de não ser possível modificar nenhum parâmetro no sistema gerido, o que acontece em muitos sistemas reais, as políticas activas limitam-se a avaliar a QoS obtida nos vários níveis de abstracção.

## 7.3 TRABALHO FUTURO

Neste trabalho propôs-se um modelo de gestão, e aplicou-se o modelo a dois cenários práticos. Tanto no modelo de gestão, como na sua aplicação prática, ficaram alguns problemas por resolver, que se apresentam nesta secção. Primeiro apresentam-se propostas para investigação futura, que correspondem a evoluções do modelo de gestão proposto, e por fim apresentam-se sugestões de melhoramentos na aplicação do modelo de gestão a casos concretos.

### 7.3.1 Áreas de Investigação Futura

**Linguagem de especificação do comportamento das políticas activas.** Seria importante definir formalmente uma linguagem para a especificação do comportamento das políticas activas para utilização em ferramentas automáticas. Esta linguagem poderia utilizar as regras propostas na secção 3.6, mas deveria ser suficientemente poderosa para permitir especificar, de forma precisa, qualquer comportamento de objectos num sistema distribuído, como é o caso de um sistema de gestão baseado em políticas activas. Espera-se que o uso de uma linguagem formal facilite o processo de refinamento do comportamento das políticas activas.

**Linguagem de especificação de contratos.** Seria também importante definir formalmente uma linguagem para a especificação de contratos para utilização em ferramentas automáticas.

**Detecção/Resolução de conflitos entre diferentes políticas.** Nos cenários apresentados, cada parâmetro sob gestão é gerido por apenas uma única política activa, de forma a evitar conflitos. Para resolver o problema genericamente, é importante estudar uma forma de detectar e resolver conflitos entre políticas activas. Algumas formas de atacar este problema foram sugeridas na secção 2.8.

**Identificação de dependências entre diferentes parâmetros.** Na metodologia apresentada, assumiu-se que se sabia qual a contribuição de cada parâmetro para a QoS. Embora a utilização de políticas activas de planeamento contribua para que não seja necessário saber a contribuição exacta de cada parâmetro, é importante estudar uma forma de avaliar, com a maior precisão possível, o que influencia cada parâmetro de QoS. Um ponto de partida possível consiste em induzir perturbações nos sistemas para identificar as dependências e os seus pesos [Brown 01].

**Avaliação da realizabilidade das garantias.** Para estabelecer contratos de fornecimento de serviço é necessário saber que garantias de QoS podem ser asseguradas, ou ter uma estimativa da probabilidade de poderem ser asseguradas. Este problema, bem como a correspondência entre os objectivos de alto nível e os parâmetros da rede e dos sistemas geridos podem ser muito complicados de resolver, exigindo normalmente um estudo caso a caso, tal como indicado na secção 3.5. A utilização de uma malha de realimentação com um integrador, tal como proposto na secção 5.5.2, pode ajudar a assegurar as garantias, sendo a saturação do integrador uma indicação de que não é possível atingir os objectivos. Embora, na realidade, seja normalmente necessário um cuidadoso planeamento da rede para que os objectivos possam ser assegurados, este assunto exige um estudo mais aprofundado para tentar identificar outras formas de avaliar se os objectivos podem ser atingidos. Um ponto de partida possível é estimar a probabilidade de limiares serem ultrapassados com base no comportamento estacionário e não estacionário dos parâmetros [Hellerstein 99].

**Falhas parciais.** No cenário de uma rede empresarial, apresentado na secção 4.5, notou-se que era possível fazer cumprir o contrato de alto nível, apesar de os contratos de níveis inferiores falharem em algumas ocasiões. Sugere-se estudar uma forma de tirar partido desta possibilidade de falhas parciais, para tornar mais eficientes as actividades de gestão.

**Modelo de preços.** Sugere-se estudar um modelo de preços [DaSilva 00] [Falkner 00], para que as políticas activas possam avaliar as suas decisões em termos de custos, imputando esses custos aos utilizadores dos serviços.

**Integração com sistemas periciais.** Sugere-se incluir na linguagem de especificação do comportamento das políticas activas a utilização de sistemas periciais, nomeadamente a utilização de lógica difusa (ver secção 3.5.1).

### 7.3.2 Melhoramentos na Aplicação Prática do Modelo

**Aplicação de especificação de contratos e construção de políticas activas.** Sugere-se o desenvolvimento de uma aplicação gráfica para facilitar a especificação de contratos, e a criação e composição de políticas activas para os diferentes níveis de abstracção. Esta aplicação deve não só facilitar a reutilização de políticas activas, como a reutilização de regras para construir novas políticas activas. Adicionalmente, a aplicação deveria detectar conflitos entre políticas activas, e permitir avaliar a realizabilidade dos contratos.

**Outras políticas de planeamento.** No cenário do fornecedor de serviço Internet do capítulo 5, apenas se desenvolveu uma política activa de planeamento para o atraso médio. Seria útil estudar como obter políticas activas de planeamento para outro tipo de requisitos nesse cenário.

**Limitações das simulações.** Deveria ser aprofundado o estudo das limitações das simulações identificadas nas secções 4.3.4 e 5.3.4. Embora se pense que as simplificações realizadas não invalidem os resultados obtidos, é sempre possível realizar simulações mais realistas. Seria também útil utilizar redes reais para avaliar na prática o funcionamento do sistema de gestão e a precisão das simulações.

**Outros cenários.** É também importante aplicar os conceitos propostos a outros cenários, pensando-se haver benefício em aplicar as técnicas propostas nesta tese a cenários com utilização de encaminhamento baseado em QoS (QoS *routing*) [RFC 2386], *Multiprotocol Label Switching* (MPLS) [RFC 2702], e cenários com utilizadores móveis.





# BIBLIOGRAFIA

- [ACL 01] Dicionário da Língua Portuguesa Contemporânea da Academia das Ciências de Lisboa. Editorial Verbo, 2001. ISBN 972-22-2046-2.
- [Al-Shaer 99] Ehab S. Al-Shaer. Programmable Agents for Active Distributed Monitoring. *Proceedings 10<sup>th</sup> IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'99)*, Zurique, Suíça, Outubro de 1999, pp.19-32.
- [Alexandrov 97] Maxim Alexandrov, Alexander Hagin, Nikolai Hagin, Yuri Karpov, Vladislav Voinov. Quality of Service management of the World Wide Web. *4<sup>th</sup> Workshop of the HP OpenView University Association, OVUA'97*, Madrid, Espanha, Abril de 1997.  
[http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/4\\_HPOVUAWS/52.ps.gz](http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/4_HPOVUAWS/52.ps.gz)
- [Anerousis 98] Nikolaos Anerousis. An Information Model for Generating Computed Views of Management Information. *DSOM'98 9<sup>th</sup> IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, Newark, DE, EUA, 26-28 de Outubro de 1998.
- [Anerousis 99] Nikolaos Anerousis. An Architecture for Building Scalable, Web-based Management Services. *Journal of Network and Systems Management*, **7**(1):73-104, Plenum Press, Março de 1999.
- [Asawa 98] Manjari Asawa. Measuring and Analyzing Service Levels: A Scalable Passive Approach. *Proceedings 6<sup>th</sup> IEEE/IFIP International Workshop on Quality of Service (IWQoS'98)*, Napa, Califórnia, EUA, 18-20 de Maio de 1998.
- [Aurrecoechea 98] Cristina Aurrecoechea, Andrew T. Campbell, Linda Hauw. A Survey of QoS Architectures. *ACM/Springer Verlag Multimedia Systems Journal*, Special Issue on QoS Architecture, **6**(3):138-151, Maio de 1998.  
<http://comet.columbia.edu/~campbell/andrew/publications/papers/multimediaa.pdf>
- [Baldi 97] Mario Baldi, Silvano Gai, Gian Pietro Picco. Exploiting Code Mobility in Decentralized and Flexible Network Management. *MA'97 First International Workshop on Mobile Agents*, Abril de 1997, pp.13-26. ISBN: 3-540-62803-7.
- [Banker 95] Kim Banker, Peter E. Mellquist. SNMP++: A Portable Object-Oriented Approach to Network Management Programming. *ConneXions*, **9**(3):35-41, Março de 1995.

- [Baritault 00] Alain Baritault. 10 Gigabit Ethernet prepara estreia. *Redes*, n.66, pp. 12, Outubro de 2000.
- [Bernardo 02] Luís Filipe Lourenço Bernardo. Um Sistema para Milhões de Utilizadores Activos com Padrões Correlacionados de Tráfego. Tese de Doutoramento. Instituto Superior Técnico. Universidade Técnica de Lisboa, Outubro de 2002.
- [Bissel 00] Torsten Bissel, Manfred Bogen, Christian Bonkowski, Volker Hadamschek. Service Level Management with Agent Technology. *Computer Networks*, **34**(2000):831-841.
- [Bhoj 01] P. Bhoj, S. Singhal, S. Chutani. SLA Management in Federated Environments. *Computer Networks*, **35** (2001):5-24.
- [Boutaba 95] Raouf Boutaba, Simon Znaty. An Architectural Approach for Integrated Network and Systems Management. *ACM Computer Communication Review*, Outubro de 1995.  
<http://www.simpleweb.org/bibliography/articles/general/zna9501.ps>
- [Brites 94] Alfredo C. S. C. Brites, Paulo A. F. Simões, Paulo M. C. Leitão, Edmundo H. S. Monteiro, Fernando P. L. Boavida Fernandes. A High-Level Notation for the Specification of Network Management Applications. *Proceedings of the INET'94, The Annual Conference of the Internet Society/JENC5, the 5<sup>th</sup> Joint European Networking Conference*, Junho de 1994, pp.561-1 a 561-8.  
<http://www.isoc.org/inet94/papers/561.ps.gz>
- [Brown 01] A. Brown, G. Kar, A. Keller. An Active Approach to Characterizing Dynamic Dependencies for Problem Determination in a Distributed Application Environment. *Proceedings of the 7<sup>th</sup> International IFIP/IEEE Symposium on Integrated Management (IM'2001)*, Seattle, EUA, 14-18 de Maio de 2001.  
[http://www.research.ibm.com/sysman/Data/Pubs/im2001\\_add.pdf](http://www.research.ibm.com/sysman/Data/Pubs/im2001_add.pdf)
- [Campbell 96] Andrew T. Campbell. A Quality of Service Architecture. PhD Thesis. Lancaster University. Janeiro de 1996.  
<http://comet.columbia.edu/~campbell/andrew/publications/papers/thesis.pdf>
- [Cardoso 02] Jorge Cardoso, Amit Sheth and John Miller. Workflow Quality of Service. *Technical Report, LSDIS Lab, Computer Science Department, University of Georgia*. Março de 2002.  
<http://lsdis.cs.uga.edu/lib/download/CSM02-TM.pdf>
- [Caruso 90] Richard E. Caruso. Network Management: A Tutorial Overview. *IEEE Communications Magazine*, **28**(3):20-25, Março de 1990.
- [Cheng 97] Nam Hong Cheng. A High-Level Approach in Network Management. *Journal of Network and Systems Management*, Plenum Publishing Corp., **5**(1):73-93, Março de 1997.

- 
- [Cherkaoui 98] Omar Cherkaoui, Abdelatif Obaid, Ahmed Serhouchni, Noemi Simoni. QOS Metrics tool using management by delegation. *Proceedings IEEE/IFIP 1998 Network Operations and Management Symposium*. Nova Orleães, EUA, 15-20 de Fevereiro de 1998, pp.836-839.
- [Chiu 89] Dah-Ming Chiu, Raj Jain. Analysis of the Increase/Decrease Algorithms for Congestion Avoidance in Computer Networks. *Journal of Computer Networks and ISDN Systems*, **17**(1):1-14, Junho de 1989.  
[http://www.cis.ohio-state.edu/~jain/papers/cong\\_av.htm](http://www.cis.ohio-state.edu/~jain/papers/cong_av.htm)
- [CIM 99] Common Information Model (CIM) Specification, Version 2.2. Distributed Management Task Force, Inc., Junho de 1999.  
[http://www.dmtf.org/download/spec/cim\\_spec\\_v22.pdf](http://www.dmtf.org/download/spec/cim_spec_v22.pdf)
- [CIMHTTP 99] Specification for CIM Operations over HTTP, Version 1.0. Distributed Management Task Force, Inc. (DMTF). 11 de Agosto de 1999.  
[http://www.dmtf.org/download/spec/xmls/CIM\\_HTTP\\_Mapping10.php](http://www.dmtf.org/download/spec/xmls/CIM_HTTP_Mapping10.php)
- [CMG] Computer Measurement Group (CMG).  
<http://cmg.org/>
- [Cochrane 91] D. Cochrane. Quality of Service (QoS). *Proceedings of the Fifth RACE TMN Conference*, 20-22 de Novembro de 1991, Londres, pp.D3IP/1/1-18.
- [CORBA 01] The Common Object Request Broker: Architecture and Specification, Revision 2.6. Object Management Group, Inc., Dezembro de 2001.  
[http://www.omg.org/technology/documents/formal/corba\\_iiop.htm](http://www.omg.org/technology/documents/formal/corba_iiop.htm)
- [CORBA TMN 00] Interworking Between CORBA and TMN Systems Specification, Version 1.0. Object Management Group, Inc., Agosto de 2000.  
<http://www.omg.org/technology/documents/formal/corba.htm>
- [Darst 99] C. Darst, S. Ramanathan. Measurement and Management of Internet Services. *Proceedings 6<sup>th</sup> IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, EUA, Maio de 1999, pp. 125-140.
- [DaSilva 00] Luiz A. DaSilva. Pricing for QoS-Enabled Networks: A Survey. *IEEE Communications Surveys*, **3**(2):2-8, 2<sup>o</sup> Trimestre de 2000.  
<http://www.comsoc.org/livepubs/surveys/public/2q00issue/dasilva.html>
- [Decisys 97] Service Level Agreements: Demonstrating how your network meets business objectives and user expectations. Decisys Inc., Julho de 1997.
- [E.800] Terms and Definitions Related to Quality of Service and Network Performance Including Dependability. Recomendação E.800 do ITU-T. Agosto de 1994.
- [E.801] Framework for Service Quality Agreement. Recomendação E.801 do ITU-T. Outubro de 1996.
-

- [Edell 99] Richard Edell, Pravin Varaiya. Providing Internet Access: What We Learn from INDEX. *IEEE Network*, **13**(5):18-25, Setembro de 1999.
- [Ensel 99] Christian Ensel. Automated generation of dependency models for service management. *HP-OpenView University Association (OVUA'99) Plenary Workshop*, Bolonha, Itália, 12-15 de Junho de 1999.  
[http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/6\\_HPOVUAWS/Papers/ensel.pdf](http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/6_HPOVUAWS/Papers/ensel.pdf)
- [Falkner 00] Matthias Falkner, Michael Devetsikiotis, Ioannis Lambadaris. An Overview of Pricing Concepts for Broadband IP Networks. *IEEE Communications Surveys*, **3**(2):2-13, 2º Trimestre de 2000.  
<http://www.comsoc.org/livepubs/surveys/public/2q00issue/falkner.html>
- [FIPS 46-3] Data Encryption Standard (DES). National Institute of Standards and Technology, U.S. Department of Commerce, Outubro de 1999.  
<http://csrc.nist.gov/publications/fips/fips46-3/fips46-3.pdf>
- [FIPS 180-1] Secure Hash Standard. National Institute of Standards and Technology, U.S. Department of Commerce, Abril de 1995.  
<http://csrc.nist.gov/publications/fips/fips180-1/fip180-1.pdf>
- [Flatin 99] Jean-Philippe Martin-Flatin, Simon Znaty, Jean-Pierre Hubaux. A Survey of Distributed Network and Systems Management Paradigms. *Journal of Network and Systems Management*, **7**(1):9-26, Plenum Press, Março de 1999.
- [Floyd 93] Sally Floyd, Van Jacobson. Random Early Detection Gateways for Congestion Avoidance. *IEEE/ACM Transactions on Networking*, **1**(4):397-413, Agosto de 1993.  
<ftp://ftp.ee.lbl.gov/papers/early.pdf>
- [Floyd 01] Sally Floyd, Vern Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, **9**(4):392-403, Agosto de 2001.  
[http://www.icir.org/floyd/papers/simulate\\_2001.pdf](http://www.icir.org/floyd/papers/simulate_2001.pdf)
- [Franceschi 96] Analúcia Schiaffino Morales De Franceschi, Luiz Fernando Kormann, Carlos Becker Westphall. Performance Evaluation for Proactive Network Management. *Proceedings ICC'96, IEEE International Conference on Communications*, 23-27, Junho 1996, Dallas, EUA, pp.22-26.
- [Franklin 87] Gene F. Franklin, J. David Powell, Abbas Emami-Naeini. Feedback Control of Dynamic Systems. Addison-Wesley, 1987. ISBN: 0-201-11540-9.
- [Franklin 87a] Secção 3.3 de [Franklin 87].
- [Franklin 87b] Secção 8.8 de [Franklin 87].
- [Gardner 96] Robert D. Gardner, David A. Harle. Methods and Systems for Alarm Correlation. *Proceedings IEEE Globecom'96*, 18-22 de Novembro de 1996, Londres, pp.136-140.

- 
- [GB 917] SLA Management Handbook. TeleManagement Forum Guide Book GB 917, Public Evaluation/Version 1.5, Junho de 2001.  
[http://www.tmfcentral.com/seguir Knowledge Base » Documents, Models & Specs » Document List » GB 917](http://www.tmfcentral.com/seguir/Knowledge%20Base%20Documents,%20Models%20&%20Specs%20Document%20List%20GB%20917)
- [GB 921] eTOM - The Business Process Framework for the Information and Communications Services Industry. TeleManagement Forum Guide Book GB 921, Public Evaluation Version 2.5, Dezembro de 2001.  
[http://www.tmfcentral.com/seguir Knowledge Base » Documents, Models & Specs » Document List » GB 921](http://www.tmfcentral.com/seguir/Knowledge%20Base%20Documents,%20Models%20&%20Specs%20Document%20List%20GB%20921)
- [Ghetie 97] Joseph G. Ghetie. Networks and Systems Management: Platforms Analysis and Evaluation. Kluwer Academic Publishers, 1997. ISBN 0-7923-9879-3.
- [Gibbens 00] R. J. Gibbens, S. K. Sargood, F. P. Kelly, H. Azmoodeh, R. Macfadyen, N. Macfadyen. An Approach to Service Level Agreements for IP networks with Differentiated Services. *Phil Trans Royal Soc Lond A*, **358**(1773):2165-2182, Agosto de 2000.  
<http://leporello.ingentaselect.com/v1=17741856/cl=25/nw=1/rpsv/cgi-bin/linker?ini=rsl&reqidx=/catchword/rs1/1364503x/v358n1773/s2/p2165>
- [Goldszmidt 93] Germán Goldszmidt, Yechiam Yemini. Evaluating Management Decisions via Delegation. *Proceedings of IFIP International Symposium on Network Management*, Abril de 1993.  
<http://www.simpleweb.org/bibliography/articles/general/gol9305.ps>  
<http://www.cs.columbia.edu/~german/papers/ifip93.ps>
- [Goldszmidt 95] Germán Goldszmidt, Yechiam Yemini. Distributed Management by Delegation. *Proceedings 15<sup>th</sup> International Conference on Distributed Computing Systems*, Junho de 1995.  
<http://www.simpleweb.org/bibliography/articles/general/gol9506.ps>
- [Goyal 99] Mukul Goyal, Arian Duresi, Padmini Misra, Chunlei Liu, Raj Jain. Effect of Number of Drop Precedences in Assured Forwarding. *GlobeCom 99*, 5-9 de Dezembro de 1999, Rio de Janeiro, Brasil.  
[http://www.cis.ohio-state.edu/~jain/papers/dpstdy\\_globecom99.htm](http://www.cis.ohio-state.edu/~jain/papers/dpstdy_globecom99.htm)
- [Haggerty 98] Paul Haggerty, Krishnan Seetharaman. The Benefits of CORBA-Based Network Management. *Communications of the ACM*, **41**(10):73-79, Outubro de 1998.
- [Hegering 99] Heinz-Gerd Hegering, Sebastian Abeck, Bernhard Neumair. Integrated Management of Networked Systems - Concepts, Architectures, and Their Operational Application. Morgan Kaufmann Publishers, 1999. ISBN: 1-55860-571-1.
- [Hegering 99a] Secção 3.4 de [Hegering 99].
-

- [Hellerstein 99] Joseph L. Hellerstein, Fan Zhang, Perwez Shahabuddin. An Approach to Predictive Detection for Service Management. *Proceedings 6<sup>th</sup> IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, EUA, Maio de 1999, pp. 309-322.
- [Ho 99] L. Lawrence Ho, Christopher J. Macey, Ronald Hiller. A Distributed and Reliable Platform for Adaptive Anomaly Detection in IP Networks. *Proceedings 10<sup>th</sup> IFIP/IEEE International Workshop on Distributed Systems: Operations and Management (DSOM'99)*, Zurique, Suíça, Outubro de 1999, pp.33-46.
- [Hood 98] Cynthia S. Hood, Chuanyi Ji. Intelligent Agents for Proactive Fault Detection. *IEEE Internet Computing*, 2(2):65-72, Março-Abril de 1998.
- [HP 99] HP OpenView - E-services Management. Hewlett-Packard Company, 1999.
- [Hurley 01] Paul Hurley, Jean-Yves Le Boudec, Patrick Thiran, Mourad Kara. ABE: Providing a Low-Delay Service within Best Effort. *IEEE Network*, 15(3):60-69, Maio/Junho de 2001.
- [JIDM 00] Inter-Domain Management: Specification & Interaction Translation. The JIDM Specification Translation and JIDM Interactive Translation Technical Standards in one volume. Open Group Technical Standard C802, Janeiro de 2000. ISBN: 1-85912-256-6.  
<http://www.opengroup.org/products/publications/catalog/c802.htm>
- [JMX] Java Management Extensions (JMX).  
<http://java.sun.com/products/JavaManagement/>
- [Kalyanasundaram 97] Pramod Kalyanasundaram, Adarshpal S. Sethi, Christopher M. Sherwin, Dong Zhu. A Spreadsheet-Based Scripting Environment for SNMP. *IM'97. Integrated Network Management V. Integrated management in a virtual world*, pp.752-765. Chapman & Hall, 1997. ISBN: 0-412-80960-5.
- [Keiser 89] Gerd E. Keiser. Local Area Networks. McGraw-Hill, 1989. ISBN: 0-07-100380-0.
- [Keiser 89a] Secção 5.4 de [Keiser 89].
- [Keiser 89b] Secção 5.5 de [Keiser 89].
- [Keshav 97] Srinivasan Keshav. An Engineering Approach to Computer Networking: ATM Networks, the Internet and the Telephone Network. Addison Wesley, 1997. ISBN: 0-201-63442-2.
- [Keshav 97a] Secção 9.4.3 de [Keshav 97].
- [Kilkki 99] Kalevi Kilkki. Differentiated Services for the Internet. Macmillan Technical Publishing, 1999. ISBN: 1-57870-132-5.
- [Kilkki 99a] Secção 6.2 de [Kilkki 99].

- 
- [Koch 96] Thomas Koch, Christoph Krell, Bernd Krämer. Policy Definition Language for Automated Management of Distributed Systems. *2<sup>nd</sup> International Workshop on Systems Management*, IEEE Computer Society, 19-21 de Junho de 1996, Canadá.  
<http://www.fernuni-hagen.de/DVT/Publikationen/Papers/pd1.ps.gz>
- [Kruse 94] R. Kruse, J. Gebhardt, F. Klawonn. Foundations of Fuzzy Systems. John Wiley & Sons, 1984. ISBN:0-471-94243- X.
- [Leiner 00] Barry M. Leiner, Vinton G. Cerf, David D. Clark, Robert E. Kahn, Leonard L. Kleinrock, Daniel C. Lynch, Jon Postel, Larry G. Roberts, Stephen Wolff. A Brief History of the Internet, version 3.31. Internet Society Document, Agosto de 2000.  
<http://www.isoc.org/internet/history/brief.shtml>
- [Lewis 97] Lundy Lewis, Jim Frey. Incorporating Business Process Management into Network and Systems Management. *Proceedings ISADS'97 - 3<sup>rd</sup> International Symposium on Autonomous Decentralised Systems*, Berlim, Alemanha, 9-11 de Abril de 1997, pp.385-392.
- [Lewis 99a] Lundy Lewis. Service Level Management of Enterprise Networks. Artech House. Setembro de 1999. ISBN: 1580530168.
- [Lewis 99b] Secção 1.2 de [Lewis 99a].
- [Lewis 99c] Capítulo 5 de [Lewis 99a].
- [Lewis 99d] Lundy Lewis. On the Integration of Service Level Management and Policy-Based Control. *Proceedings Policy Workshop'99 (Policy'99)*, Bristol, UK, 15-19 de Novembro de 1999.  
<http://www-dse.doc.ic.ac.uk/events/policy-99/pdf/14b-Lewis.pdf>
- [Lewis 02] Lundy Lewis, Pradeep Ray. On the Migration from Enterprise Management to Integrated Service Level Management. *IEEE Network*, **16**(1):8-14, Janeiro/Fevereiro de 2002.
- [Leydekkers 96] Peter Leydekkers, Valérie Gay. ODP View on Quality of Service for Open Distributed Multimedia Environments. *4<sup>th</sup> IFIP Workshop on QoS (IWQoS'96)*, Paris, França, Março de 1996.
- [Lin 98] Along Lin. A Hybrid Approach to Fault Diagnosis in Network and System Management. Hewlett Packard Technical Report HPL-98-20. Fevereiro de 1998.  
<http://www.hpl.hp.com/techreports/98/HPL-98-20.pdf>
- [Liu 01] Karen Liu, John Ryan. All the Animals in the Zoo: The Expanding Menagerie of Optical Components. *IEEE Communications Magazine*, **39**(7):110-115, Julho de 2001.
- [Lippmann 87] Richard P. Lippmann. An Introduction to Computing with Neural Nets. *IEEE ASSP Magazine*, Abril de 1987, pp.4-22.
-

- [Lupu 95] Emil C. Lupu, Damian A. Marriott, Morris S. Sloman, Nicholas Yialelis. A Policy Based Role Framework for Access Control. *First ACM/NIST Role Based Access Control Workshop*, Gaithersburg, EUA, Dezembro de 1995.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/rbac95.ps.z>
- [Lupu 97a] Emil C. Lupu, Morris Sloman. Towards a Role Based Framework for Distributed Systems Management. *Journal of Network and Systems Management*, **5**(1):5-30, Plenum Press, 1997.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/jnsm96.ps.z>
- [Lupu 97b] Emil Lupu, Morris Sloman. Reconciling Role Based Management and Role Based Access Control. *ACM/NIST Role Based Access Control Workshop*, Virgínia, EUA, 6-7 de Novembro de 1997.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/RBAC97.PS.GZ>
- [Lupu 98] Emil C. Lupu. A Role-Based Framework for Distributed Systems Management. PhD Thesis. Imperial College, Londres. Julho de 1998.  
<http://www.doc.ic.ac.uk/~ecl1/thesis/>
- [Lupu 99] Emil Lupu, Morris Sloman. Conflicts in Policy-based Distributed Systems Management, *IEEE Transactions on Software Engineering - Special Issue on Inconsistency Management*, **25**(6):852-869, Novembro/Dezembro de 1999.  
<http://www.doc.ic.ac.uk/~ecl1/papers/tse.pdf>
- [M.3010] Principles for a Telecommunications Management Network. Recomendação M.3010 do ITU-T. Maio de 1996.
- [M.3400] TMN Management Functions. Recomendação M.3400 do ITU-T. Abril de 1997.
- [Mampaey 00] Marcel Mampaey, Alban Couturier. Using TINA Concepts for IN Evolution. *IEEE Communications Magazine*, **38**(6):94-99, Junho de 2000.
- [Marriott 96] Damian A. Marriott, Morris S. Sloman. Management Policy Service for Distributed Systems. *Proceedings of IEEE 3<sup>rd</sup> International Workshop on Services in Distributed and Networked Environments (SDNE'96)*, Macau, Junho de 1996, IEEE Computer Society Press.  
[ftp://dse.doc.ic.ac.uk/dse-papers/management/policy\\_service\\_sдне96.pdf](ftp://dse.doc.ic.ac.uk/dse-papers/management/policy_service_sдне96.pdf)
- [McBride] Service Level Agreements: A Path to Effective System Management and Application Optimization. White paper, Hewlett-Packard Company.
- [Meira 97] Dilmar Malheiros Meira, José Marcos S. Nogueira. Métodos e Algoritmos para Correlação de Alarmes em Redes de Telecomunicações. *Proceedings SBRC'97, 15<sup>o</sup> Simpósio Brasileiro de Redes de Computadores*, 1997, pp.79-98.



- 
- [Meyer 95] Kraig Meyer, Mike Erlinger, Joe Betser, Carl Sunshine, Germán Goldszmidt, Yechiam Yemini. Decentralizing Control and Intelligence in Network Management. *Proceedings 4<sup>th</sup> International Symposium on Integrated Network Management*, Santa Bárbara, CA, EUA, Maio de 1995.  
<http://www.simpleweb.org/bibliography/articles/general/gol9505.ps>
- [Mitrani 82] I. Mitrani. *Simulation Techniques for Discrete Event Systems*. Cambridge University Press, 1982. ISBN 0-521-28282-9.
- [Moffett 93a] Jonathan D. Moffett, Morris S. Sloman. Policy Conflict Analysis in Distributed System Management. *Journal of Organizational Computing*, Vol. 4, n. 1, 1993.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/conflict.ps.z>
- [Moffett 93b] Jonathan D. Moffett, Morris S. Sloman. Policy Hierarchies for Distributed Systems Management. *IEEE Journal on Selected Areas in Communications, Special Issue on Network Management*, **11**(9):1404-1414, Dezembro de 1993.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/policyhierarchy.ps.z>
- [NET-SNMP] The NET-SNMP Project. Previamente conhecido como: The University of California at Davis SNMP Project.  
<http://net-snmp.sourceforge.net/>
- [NS] UCB/LBNL/USC/VINT Network Simulator (version 2).  
<http://www.isi.edu/nsnam/ns/>
- [NS-DiffServ] Sean Murphy. DiffServ additions to NS.  
<http://www.teltec.dcu.ie/~murphys/ns-work/diffserv/index.html>
- [Orfali 96] Robert Orfali, Dan Harkey, Jeri Edwards. *The Essential Distributed Objects Survival Guide*. John Wiley&Sons, Inc. 1996. ISBN: 0471-12993-3.
- [Orfali 96a] Capítulo 13 de [Orfali 96].
- [Ousterhout 94] John K. Ousterhout. *Tcl and the Tk Toolkit*. Addison-Wesley, 1994. ISBN: 0-201-63337-X.
- [Parekh 01] Sujay Parekh, Neha Gandhi, Joe Hellerstein, Dawn Tilbury, T. S. Jayram, Joe Bigus. Using Control Theory to Achieve Service Level Objectives in Performance Management. *7<sup>th</sup> IFIP/IEEE International Symposium on Integrated Network Management (IM'2001)*, Seattle, EUA, 14-18 de Maio de 2001.
- [Pavón 98] Juan Pavón, José Tomás. CORBA for Network and Service Management in the TINA Framework. *IEEE Communications Magazine*, **36**(3):72-79, Março de 1998.
- [Paxson 95] Vern Paxson, Sally Floyd. Wide Area Traffic: The Failure of Poisson Modeling. *IEEE/ACM Transactions on Networking*, **3**(3):226-244, Junho 1995.  
<http://www.icir.org/floyd/papers/WAN-poisson.ps.z>
-

- [Pereira 98] Paulo Pereira, Jorge Sousa, Pedro Teixeira, Paulo Pinto. Plataforma Dinâmica de Gestão de Redes. *Actas da 1ª Conferência sobre Redes de Computadores, CRC'98*, Coimbra, Portugal, 9-10 de Novembro de 1998, pp.31-34.  
<http://www.uc.pt/crc98/comfin05/comfin05.pdf>  
<http://mariel.inesc.pt/~prbp/publica/crc98/crc98.html>
- [Pereira 99a] Paulo Pereira, Pedro Pessoa, Luís Agostinho, Paulo Pinto. Gestão de Redes e Sistemas por Scripts. *Actas da 2ª Conferência sobre Redes de Computadores, CRC'99*, Évora, Portugal, 18-19 de Outubro de 1999.  
<http://mariel.inesc.pt/~prbp/publica/crc99/crc99.htm>
- [Pereira 99b] Paulo Pereira, Paulo Pinto. Algorithms and Contracts for Network and Systems Management. *IEEE LANOMS'99, Latin American Network Operations and Management Symposium*, Rio de Janeiro, Brasil, 3-5 de Dezembro de 1999, pp. 385-396. ISBN: 85-900382-3-8.  
<http://mariel.inesc.pt/~prbp/publica/lanoms99.pdf>
- [Pereira 00a] Paulo Pereira, Elionildo Menezes, Djamel Sadok e Paulo Pinto. Management of Differentiated Services with Active Policies. *4th International Student Conference on Electrical Engineering, POSTER'2000*, pp. C13, 25 de Maio de 2000, Faculdade de Engenharia Electrotécnica, Universidade Técnica de Praga, República Checa.  
<http://mariel.inesc.pt/~prbp/publica/poster2000.pdf>
- [Pereira 00b] Elionildo Menezes, Djamel Sadok, Paulo Pereira. Uma Abordagem para Implementação de Gerenciamento de Políticas em Redes de Serviços Diferenciados. *V Workshop TMN do 18º Simpósio Brasileiro de Redes de Computadores (SBRC'2000)*. Belo Horizonte, MG, Brasil, 25 de Maio de 2000.  
<http://mariel.inesc.pt/~prbp/publica/wtmn.pdf>
- [Pereira 00c] Elionildo Menezes, Djamel Sadok, Judith Kelner, Paulo Pereira, Paulo Pinto. Service Management for Differentiated Services Networks. *IEEE Workshop on IP-Oriented Operations & Management, IPOM'2000*, Cracóvia, Polónia, 4-6 de Setembro de 2000, pp. 99-108. ISBN: 83-88309-00-5.  
<http://mariel.inesc.pt/~prbp/publica/ipom2000.pdf>
- [Pereira 01a] Paulo Pereira, Djamel Sadok, Paulo Pinto. Service Level Management of Differentiated Services Networks with Active Policies. *3rd Conference on Telecommunications, ConfTele'2001*, Figueira da Foz, Portugal, 23-24 de Abril de 2001, pp.542-546. ISBN: 972-98115-2-0 (livro), 972-98115-3-9 (CDROM).  
<http://mariel.inesc.pt/~prbp/publica/conftele2001slm.pdf>
- [Pereira 01b] Paulo Pereira, Bruno Afonso, Daniel Gomes. Differentiated Services Network Simulation. *3rd Conference on Telecommunications, ConfTele'2001*, Figueira da Foz, Portugal, 23-24 de Abril de 2001, pp.562-566. ISBN: 972-98115-2-0 (livro), 972-98115-3-9 (CDROM).  
<http://mariel.inesc.pt/~prbp/publica/conftele2001ns.pdf>

- 
- [Peterson 96] Larry L. Peterson, Bruce S. Davie. *Computer Networks - A Systems Approach*. Morgan Kaufman, 1996. ISBN: 1-55860-404-9.
- [Peterson 96a] Seção 8.2.2 de [Peterson 96].
- [Post 96] Michael Post, Chien-Chung Shen, John Wei. The Manager/Agency Paradigm for Distributed Network Management. *NOMS'96, IEEE Network Operations and Management Symposium*, Abril de 1996, pp. 44-53. ISBN: 0-7803-2518-4.
- [Pozzi 96] Michael Pozzi. Proactive Network Analysis Techniques. *The Enterprise Network Management and Troubleshooting Symposium*. Hewlett Packard, Outubro de 1996, pp.81-112.
- [Pritsker 86] A. Alan B. Pritsker. *Introduction to Simulation and SLAM II*, 3<sup>rd</sup> Edition. Halsted Press, John Wiley & Sons, 1986. ISBN 0-470-20292-0.
- [Psounis 99] Konstantinos Psounis. Active Networks: Applications, Security, Safety, and Architectures. *IEEE Communications Surveys*, 1<sup>o</sup> Trimestre de 1999, pp.2-16.  
<http://www.comsoc.org/livepubs/surveys/public/1q99issue/psounis.html>
- [Ptolemy] Ptolemy Project Home Page.  
<http://ptolemy.eecs.berkeley.edu/>
- [Puka 99] Daniel Puka, Manoel Cammillo Penna, Vinicius Prodocimo. Service Level Management in ATM Networks. *IEEE LANOMS'99, Latin American Network Operations and Management Symposium*, Rio de Janeiro, Brasil, 3-5 de Dezembro de 1999, pp.179-190.
- [QBone] QBone Home Page.  
<http://qbone.internet2.edu/>
- [Raman 98] Lakshmi Raman. OSI Systems and Network Management. *IEEE Communications Magazine*, **36**(3):46-53, Março de 1998.
- [Ramanathan 01] Ananthanarayanan Ramanathan, Manish Parashar. Active Resource Management for The Differentiated Services Environment. *Active Middleware Services Workshop*, Kluwer Academic Publishers, Agosto de 2001.  
<http://www.caip.rutgers.edu/TASSL/Papers/aqos-ams01.pdf>
- [RFC 792] J. Postel. Internet Control Message Protocol. IETF STD 5, RFC 792. Setembro de 1981.
- [RFC 1155] Marshall Rose, Keith McCloghrie. Structure and Identification of Management Information for TCP/IP-based Internets. IETF STD 16, RFC 1155. Maio de 1990.
- [RFC 1157] J. Case, M. Fedor, M. Schoffstall, J. Davin. A Simple Network Management Protocol (SNMP). IETF STD 15, RFC 1157. Maio de 1990.
- [RFC 1212] Marshall Rose, Keith McCloghrie. Concise MIB Definitions. IETF STD 16, RFC 1212. Março de 1991.
-

- [RFC 1213] Keith McCloghrie, Marshall Rose. Management Information Base for Network Management of TCP/IP-based internets: MIB-II. IETF STD 17, RFC 1213. Março de 1991.
- [RFC 1321] R. Rivest. The MD5 Message-Digest Algorithm. IETF RFC 1321. Abril de 1992.
- [RFC 1633] R. Braden, D. Clark, S. Shenker. Integrated Services in the Internet Architecture: An Overview. IETF RFC 1633. Junho de 1994.
- [RFC 1905] J. Case, K. McCloghrie, M. Rose, S. Waldbusser. Protocol Operations for Version 2 of the Simple Network Management Protocol (SNMPv2), IETF RFC 1905. Janeiro de 1996.
- [RFC 1909] Keith McCloghrie. An Administrative Infrastructure for SNMPv2. IETF RFC 1909. Fevereiro de 1996.
- [RFC 1990] K. Sklower, B. Lloyd, G. McGregor, D. Carr, T. Coradetti. The PPP Multilink Protocol (MP). IETF RFC 1990. Agosto de 1996.
- [RFC 2021] S. Waldbusser. Remote Network Monitoring Management Information Base Version 2 using SMIV2. IETF RFC 2021. Janeiro de 1997.
- [RFC 2205] R. Braden. Resource ReSerVation Protocol (RSVP) - Version 1 Functional Specification. IETF RFC 2205. Setembro de 1997.
- [RFC 2211] J. Wroclawski. Specification of the Controlled-Load Network Element Service. IETF RFC 2211. Setembro de 1997.
- [RFC 2212] S. Shenker, C. Partridge, R. Guerin. Specification of Guaranteed Quality of Service. IETF RFC 2212. Setembro de 1997.
- [RFC 2330] V. Paxson, G. Almes, J. Mahdavi, M. Mathis. Framework for IP Performance Metrics. IETF RFC 2330. Maio de 1998.
- [RFC 2386] E. Crawley, R. Nair, B. Rajagopalan, H. Sandick. A Framework for QoS-based Routing in the Internet. IETF RFC 2386. Agosto de 1998.
- [RFC 2474] K. Nichols, S. Blake, F. Baker, D. Black. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474. Dezembro de 1998.
- [RFC 2475] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss. An Architecture for Differentiated Services. IETF RFC 2475. Dezembro de 1998.
- [RFC 2578] K. McCloghrie, D. Perkins, J. Schoenwaelder. Structure of Management Information Version 2 (SMIV2). IETF STD 58, RFC 2578. Abril de 1999.
- [RFC 2597] J. Heinanen, F. Baker, W. Weiss, J. Wroclawski. Assured Forwarding PHB Group. IETF RFC 2597. Junho de 1999.

- 
- [RFC 2598] V. Jacobson, K. Nichols, K. Poduri. An Expedited Forwarding PHB. IETF RFC 2598. Junho de 1999.
- [RFC 2638] K. Nichols, V. Jacobson, L. Zhang. A Two-bit Differentiated Services Architecture for the Internet. IETF RFC 2638. Julho de 1999.
- [RFC 2698] J. Heinanen, R. Guerin. A Two Rate Three Color Marker. IETF RFC 2698. Setembro de 1999.
- [RFC 2702] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, J. McManus. Requirements for Traffic Engineering Over MPLS. IETF RFC 2702. Setembro de 1999.
- [RFC 2748] J. Boyle, R. Cohen, S. Herzog, R. Rajan, A. Sastry. The COPS (Common Open Policy Service) Protocol. IETF RFC 2748. Janeiro de 2000.
- [RFC 2753] R. Yavatkar, D. Pendarakis, R. Guerin. A Framework for Policy-based Admission Control. IETF RFC 2753. Janeiro de 2000.
- [RFC 2819] S. Waldbusser. Remote Network Monitoring Management Information Base. IETF STD 59, RFC 2819. Maio de 2000.
- [RFC 2981] R. Kavasseri, B. Stewart. Event MIB. IETF RFC 2981. Outubro de 2000.
- [RFC 2982] R. Kavasseri, B. Stewart. Distributed Management Expression MIB. IETF RFC 2982. Outubro de 2000.
- [RFC 3060] B. Moore, E. Ellesson, J. Strassner, A. Westerinen. Policy Core Information Model - Version 1 Specification. IETF RFC 3060. Fevereiro de 2001.
- [RFC 3084] K. Chan, J. Seligson, D. Durham, S. Gai, K. McCloghrie, S. Herzog, F. Reichmeyer, R. Yavatkar, A. Smith. COPS Usage for Policy Provisioning (COPS-PR). IETF RFC 3084. Março de 2001.
- [RFC 3165] D. Levi, J. Schoenwaelder. Definitions of Managed Objects for the Delegation of Management Scripts. IETF RFC 3165. Agosto de 2001.
- [RFC 3174] D. Eastlake, 3rd, P. Jones. US Secure Hash Algorithm 1 (SHA1). IETF RFC 3174. Setembro de 2001.
- [RFC 3198] A. Westerinen, J. Schnizlein, J. Strassner, M. Scherling, B. Quinn, S. Herzog, A. Huynh, M. Carlson, J. Perry, S. Waldbusser. Terminology for Policy-Based Management. IETF RFC 3198. Novembro de 2001.
- [RFC 3231] D. Levi, J. Schoenwaelder. Definitions of Managed Objects for Scheduling Management Operations. IETF RFC 3231. Janeiro de 2002.
- [RFC 3411] D. Harrington, R. Presuhn, B. Wijnen. An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks. IETF STD 62, RFC 3411. Dezembro de 2002.
-

- [RSA 78] R. L. Rivest, A. Shamir, L. Adleman. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. *Communications of the ACM*, **21**(2):120-126, Fevereiro de 1978.
- [Ryan 98] John P. Ryan. WDM: North American Deployment Trends. *IEEE Communications Magazine*, **36**(2):40-44, Fevereiro de 1998.
- [Sahai 97] Akhil Sahai, Stéphane Billiard, Christine Morin. Astrolog: A Distributed and Dynamic Environment for Network and System Management. *Proceedings of the 1<sup>st</sup> European Information Infrastructure User Conference*, Estugarda, Alemanha, Fevereiro de 1997.  
<http://www.irisa.fr/solidor/doc/ps97/infra.ps.gz>
- [Schmidt 00] Holger Schmidt. Service Level Agreements based on Business Process Modeling. *HP-OpenView University Association (OVUA'00) Plenary Workshop*, Santorini, Grécia, Junho de 2000.  
[http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/7\\_HPOVUAWS/papers/session3-2.pdf](http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/7_HPOVUAWS/papers/session3-2.pdf)
- [Schonwalder 95] Jürgen Schönwälder, H. Langendörfer. Tcl Extensions for Network Management Applications. *3<sup>rd</sup> Tcl/Tk Workshop*, Toronto, Julho de 1995, pp.279-288.  
<http://www.ibr.cs.tu-bs.de/users/schoenw/papers/tcltk-95.ps.gz>
- [Schonwalder 96] Jürgen Schönwälder. Using Multicast-SNMP to Coordinate Distributed Management Agents. *2<sup>nd</sup> IEEE Workshop on Systems Management*, Toronto, 20 de Junho de 1996.  
<http://www.ibr.cs.tu-bs.de/users/schoenw/papers/dsm-96.ps.gz>
- [Siegl 96] Manfred R. Siegl, Georg Trausmuth. Hierarchical Network Management, A Concept and its Prototype in SNMPv2. *Computer Networks and ISDN Systems*, **28**(1996):441-452.
- [Sloman 94] Morris Sloman. Policy Driven Management for Distributed Systems. *Journal of Network and Systems Management*, Plenum Press. **2**(4):333-360, Dezembro de 1994.  
<ftp://dse.doc.ic.ac.uk/dse-papers/management/pdman.ps.Z>
- [Stadler 96] Michael Stadler. Mapping Management Information to Service Interfaces Supporting High-Level Network Management Applications. *DSOM'96, 7<sup>th</sup> IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, L'Aquila, Itália, 28-30 de Outubro de 1996.  
<http://einstein.informatik.uni-oldenburg.de/papers/mapping.ps.gz>
- [Stamatelopoulos 95] F. Stamatelopoulos, T. Chiotis, B. Maglaris. A Scaleable, Platform-Based Architecture for Multiple Domain Network Management. *Proceedings IEEE ICC'95*, 18-22 de Junho de 1995, Seattle, EUA, pp.1453-1458.

- 
- [Stamatelopoulos 97] F. Stamatelopoulos, B. Maglaris. A Management Architecture for Internet Information Services. *4<sup>th</sup> Workshop of the HP OpenView University Association, OVUA'97*, April 1997, Madrid, Espanha.  
[http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/4\\_HPOVUAWS/53.ps.gz](http://www.hpovua.org/PUBLICATIONS/PROCEEDINGS/4_HPOVUAWS/53.ps.gz)
- [Stallings 99] William Stallings. *SNMP, SNMPv2, SNMPv3 and RMON 1 and 2*, 3<sup>rd</sup> Edition. Addison Wesley. 1999. ISBN: 0-201-48534-6.
- [Sturm 00] Rick Sturm, Wayne Morris, Mary Jander. *Foundations of Service Level Management*. SAMS, Abril de 2000. ISBN:0-672-31743-5.
- [Tanenbaum 81] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 1981. ISBN: 0-13-165183-8.
- [Tanenbaum 81a] Secção 2.2.2 de [Tanenbaum 81].
- [Teitelbaum 99] Benjamin Teitelbaum, Susan Hares, Larry Dunn, Robert Neilson, Vishy Narayan, Francis Reichmeyer. Internet2 QBone: Building a Testbed for Differentiated Services. *IEEE Network*, **13**(5):8-16, Setembro de 1999.
- [Tennenhouse 97] David L. Tennenhouse, Jonathan M. Smith, W. David Sincoskie, David J. Wetherall, Gary J. Minden. A Survey of Active Network Research. *IEEE Communications Magazine*, **35**(1):80-86, Janeiro de 1997.
- [Tequila] Projecto IST-Tequila. Traffic Engineering for Quality of Service in the Internet, at Large Scale.  
<http://www.ist-tequila.org/>
- [Thottan 98] Marina Thottan, Chuanyi Ji. Proactive Anomaly Detection Using Distributed Intelligent Agents. *IEEE Network*, **12**(5):21-27, Setembro-Outubro 1998.
- [Tivoli] Tivoli Systems Management.  
<http://www.tivoli.com/>
- [Toh 02] C.-K. Toh. *Ad Hoc Mobile Wireless Networks: Protocols and Systems*. Prentice Hall, 2002. ISBN: 9-13-007817-4.
- [Trimintzios 01] P. Trimintzios, I. Andrikopoulos, G. Pavlou, C. F. Cavalcanti, D. Goderis, Y. T'Joens, P. Georgatsos, L. Georgiadis, D. Griffin, C. Jacquenet, R. Egan, G. Memenios. An Architectural Framework for Providing QoS in IP Differentiated Services Networks. *7<sup>th</sup> IFIP/IEEE International Symposium on Integrated Network Management (IM'2001)*, Seattle, EUA, 14-18 de Maio de 2001.  
<http://www.ist-tequila.org/publications/tequila-IM-2001.pdf>
- [UUNET] UUNET / WorldCom / MCI. Internet Service Level Agreement.  
<http://global.mci.com/terms/sla/>
- [Valimaa 95] Harri Valimaa, Tapani Honkanen. Producing Quality Factors of LAN Interconnection Services. *Proceedings INET'95*.  
<http://www.isoc.org/HMP/PAPER/069/ps/paper.ps>
-

- [Vieira 96] Elvis Melo Vieira, Solange Teresinha Sari. Prototipação de um Subagente Adaptativo Baseado em Redes Neurais. *Proceedings SBRC'96, 14º Simpósio Brasileiro de Redes de Computadores*, Fortaleza, Brasil, Maio de 1996, pp.140-156.
- [Vinoski 02] Steve Vinoski. Where is Middleware? *IEEE Internet Computing*, **6**(2):83-85, Março-Abril de 2002.
- [WBEM] Web-Based Enterprise Management (WBEM) Initiative.  
[http://www.dmtf.org/standards/standard\\_wbem.php](http://www.dmtf.org/standards/standard_wbem.php)
- [Wetherall 95] David Wetherall, Christopher J. Lindblad. Extending Tcl for Dynamic Object-Oriented Programming. *Proceedings of the Tcl/Tk Workshop '95*, Toronto, Julho de 1995.  
<http://www.tns.lcs.mit.edu/publications/tcltk95.djw.html>
- [Wies 94a] René Wies. Policies in Network and Systems Management – Formal Definition and Architecture. *Journal of Network and Systems Management*, Plenum Publishing Corp., **2**(1):63-83, Março de 1994.  
<http://www.simpleweb.org/bibliography/articles/general/wie9506.ps>
- [Wies 94b] René Wies. Policy Definition and Classification: Aspects, Criteria, and Examples. *Proceedings of the IFIP/IEEE International Workshop on Distributed Systems: Operations & Management*, Toulouse, França, 10-12 de Outubro de 1994.  
<http://www.simpleweb.org/bibliography/articles/general/wie9505.ps>
- [WinSNMP] Windows SNMP API.  
<ftp://sunsite.unc.edu/pub/micro/pc-stuff/ms-windows/winsnmp/>
- [X.208] Specification of Abstract Syntax Notation One (ASN.1). Recomendação X.208 do ITU-T. 1988.
- [X.209] Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). Recomendação X.209 do ITU-T. 1988.
- [X.641] Quality of Service: Framework. Recomendação X.641 do ITU-T. Dezembro de 1997.
- [X.700] Management Framework for Open Systems Interconnection (OSI) for CCITT Applications. Recomendação X.700 do ITU-T. Setembro de 1992.
- [X.701] Systems Management Overview. Recomendação X.701 do ITU-T. Agosto de 1997.
- [X.710] Common Management Information Service. Recomendação X.710 do ITU-T. Outubro de 1997.
- [X.711] Common Management Information Protocol: Specification. Recomendação X.711 do ITU-T. Outubro de 1997.
- [X.720] Structure of Management Information: Management Information Model. Recomendação X.720 do ITU-T. Janeiro de 1992.



- 
- [X.721] Structure of Management Information: Definition of Management Information. Recomendação X.721 do ITU-T. Fevereiro de 1992.
- [X.722] Structure of Management Information: Guidelines for Definition of Managed Objects. Recomendação X.722 do ITU-T. Janeiro de 1992.
- [X.733] Systems Management: Alarm Reporting Function. Recomendação X.733 do ITU-T. Fevereiro de 1992.
- [X.739] Systems Management: Metric Objects and Attributes. Recomendação X.739 do ITU-T. Novembro de 1993.
- [X.749] Systems Management: Management Domain and Management Policy Management Function. Recomendação X.749 do ITU-T. Agosto de 1997.
- [X.753] Systems Management: Command Sequencer for Systems Management. Recomendação X.753 do ITU-T. Outubro de 1997.
- [X.901] Open Distributed Processing - Reference Model: Overview. Recomendação X.901 do ITU-T. Agosto de 1997.
- [X.902] Open Distributed Processing - Reference Model: Foundations. Recomendação X.902 do ITU-T. Novembro de 1995.
- [X.903] Open Distributed Processing - Reference Model: Architecture. Recomendação X.903 do ITU-T. Novembro de 1995.
- [X.906draft] Open Distributed Processing - Reference Model: Quality of Service. ISO/IEC JTC 1/SC 33 N 145. Rascunho da Recomendação X.906 do ITU-T. Janeiro de 1998.
- [X.911draft] Information Technology - Open Distributed Processing - Reference Model - Enterprise Viewpoint. ISO/IEC JTC1/SC7/WG3 3N34. Rascunho da Recomendação X.911 do ITU-T. Setembro de 1999.
- [XML 00] Extensible Markup Language (XML) 1.0 (Second Edition). W3C Recommendation, 6 de Outubro de 2000.  
<http://www.w3.org/TR/REC-xml>
- [XMLCIM 99] Specification for the Representation of CIM in XML, Version 2.0. Distributed Management Task Force, Inc. (DMTF). 20 de Julho de 1999.  
[http://www.dmtf.org/download/spec/xmls/CIM\\_XML\\_Mapping20.php](http://www.dmtf.org/download/spec/xmls/CIM_XML_Mapping20.php)
- [Y.1541] Internet Protocol Aspects – Quality of Service and Network Performance: Network Performance Objectives for IP-Based Services. Pré-publicação da Recomendação Y.1541 do ITU-T. Maio de 2002.
- [Yemini 96] S. Yemini, S. Klinger, E. Mozes, Y. Yemini, D. Ohsie. High Speed & Robust Event Correlation. *IEEE Communications Magazine*, **34**(5):82-90, Maio de 1996.
-

- [Yoshihara 99] K. Yoshihara, K. Sugiyama, H. Horiuchi, S. Obana. Dynamic Polling Scheme Based on Time Variation of Network Management Information Values. *Proceedings 6<sup>th</sup> IFIP/IEEE International Symposium on Integrated Network Management (IM'99)*, Boston, MA, EUA, Maio de 1999, pp. 141-154.
- [Znaty 94] Simon Znaty, Jean Sclavos. Annotated Bibliography on Network Management. *ACM Computer Communication Review*, **24**(1):37-56, 1994.
- [Znaty 96] Simon Znaty, Michel Lion, Jean-Pierre Hubaux. DEAL: A DElegated Agent Language for Developing Network Management Functions. *PAAM'96 The first International Conference and Exhibition on the Pratical Application of Intelligent Agents and Multi-Agent Technology*, Londres, Abril de 1996.

# APÊNDICE A

## LISTA DE SIGLAS

|       |  |
|-------|--|
| ABE   | <i>Alternative Best Effort.</i>  |
| AF    | <i>Assured Forwarding.</i> Envio Assegurado.   |
| API   | <i>Application Programming Interface.</i> Interface de programação de aplicações.  |
| ASN.1 | <i>Abstract Syntax Notation One.</i>   |
| ATM   | <i>Asynchronous Transfer Mode.</i> Tecnologia de comutação e multiplexagem de alto débito que pode transmitir diversos tipos de tráfego utilizando pacotes de tamanho fixo chamados células. |
| BB    | <i>Bandwidth Broker.</i> Corretor de Largura de Banda.   |
| BE    | <i>Best Effort.</i> Melhor Esforço.  |
| BOD   | <i>Bandwidth on Demand.</i> Largura de banda a pedido.   |
| bps   | bit por segundo.   |
| CBR   | <i>Constant Bit Rate.</i> Ritmo binário constante.   |
| CIM   | <i>Common Information Model.</i>   |
| CLA   | <i>Customer Level Agreement.</i> Acordo de utilizador.   |
| CMG   | <i>Computer Measurement Group.</i>   |
| CMIP  | <i>Common Management Information Protocol.</i> O protocolo OSI para gestão de redes.   |
| CMIS  | <i>Common Management Information Service.</i> O serviço oferecido pelo protocolo CMIP.   |
| CMISE | <i>Common Management Information Service Element.</i> O elemento de serviço responsável pela troca de informação de gestão de redes.   |
| COPS  | <i>Common Open Policy Service.</i>   |
| CORBA | <i>Common Object Request Broker Architecture.</i>  |
| DES   | <i>Data Encryption Standard.</i>   |
| DEN   | <i>Directory Enabled Networks.</i>   |

|       |  |
|-------|--|
| DMTF  | <i>Distributed Management Task Force.</i>  |
| DSCP  | <i>Differentiated Services Codepoint.</i> Código de serviço diferenciado.  |
| DSL   | <i>Digital Subscriber Line.</i>  |
| EF    | <i>Expedited Forwarding.</i> Envio expedito.   |
| EWMA  | <i>Exponential Weighted Moving Average.</i> Média móvel de peso exponencial.   |
| FCAPS | <i>Fault, Configuration, Accounting, Performance and Security.</i> Falhas, Configuração, Contabilização, Desempenho e Segurança.                   |
| FTP   | <i>File Transfer Protocol.</i> Protocolo Internet de nível aplicação que permite transferir ficheiros entre máquinas recorrendo a TCP/IP.          |
| GDMO  | <i>Guidelines for Definition of Managed Objects.</i>   |
| HTML  | <i>Hypertext Markup Language.</i> Linguagem que permite criar documentos hipertexto utilizados em servidores WWW.                                  |
| HTTP  | <i>Hypertext Transfer Protocol.</i> O protocolo que é utilizado para transferir documentos HTML entre servidores e cliente WWW.                    |
| HTTPS | HTTP seguro. Versão do HTTP com dados cifrados.  |
| ICMP  | <i>Internet Control Message Protocol.</i>  |
| IEEE  | <i>Institute of Electrical and Electronics Engineers.</i>  |
| IETF  | <i>Internet Engineering Task Force.</i>  |
| IP    | <i>Internet Protocol.</i> O protocolo de nível rede utilizado na Internet.   |
| IPng  | <i>IP next generation.</i> IPv6.   |
| IPv4  | IP versão 4.   |
| IPv6  | IP versão 6.   |
| ISDN  | <i>Integrated Services Digital Network.</i> Rede Digital com Integração de Serviços.   |
| ISP   | <i>Internet Service Provider.</i> Fornecedor de serviço Internet.  |
| ISO   | <i>International Organization for Standardization.</i>   |
| ITU   | <i>International Telecommunications Union.</i> Organização de normalização compreendendo as administrações postais e telefónicas de vários países. |
| ITU-T | <i>ITU - Telecommunication Standardization Sector.</i>   |
| JIDM  | <i>Joint Inter-Domain Management.</i>  |
| LDAP  | <i>Lightweight Directory Access Protocol.</i>  |
| LLA   | <i>Logical Layered Architecture.</i> Arquitectura de Camadas Lógicas. Arquitectura de níveis de TMN.   |

|      |  |
|------|--|
| MBD  | <i>Management by Delegation.</i> Gestão por Delegação.   |
| MD5  | <i>Message Digest 5.</i>   |
| MIB  | <i>Management Information Base.</i> Base de dados de informação de gestão. Um conjunto estruturado de objectos que representam recursos a ser geridos.   |
| MIT  | Massachusetts Institute of Technology.   |
| MOF  | <i>Managed Object Format.</i>  |
| MPLS | <i>Multiprotocol Label Switching.</i>  |
| nam  | <i>Network Animator.</i>   |
| NNTP | <i>Network News Transfer Protocol.</i>   |
| NP   | <i>Network Performance.</i> Qualidade de funcionamento da rede.  |
| ns   | <i>Network Simulator.</i>  |
| NTP  | <i>Network Time Protocol.</i>  |
| ODP  | <i>Open Distributed Processing.</i> Processamento distribuído aberto.  |
| OMG  | <i>Object Management Group.</i>  |
| OSI  | <i>Open Systems Interconnection.</i> Um esforço internacional para facilitar as comunicações entre computadores de diferentes fabricantes e tecnologias. |
| OTcl | <i>MIT Object Tool Command Language.</i> Uma versão orientada para objectos da linguagem Tcl desenvolvida pelo MIT.                                      |
| PDU  | <i>Protocol Data Unit.</i> Unidade de dados do protocolo.  |
| PDP  | <i>Policy Decision Point.</i> Ponto de decisão de políticas.   |
| PEP  | <i>Policy Enforcement Point.</i> Ponto de execução de políticas.   |
| PHB  | <i>Per-Hop Behaviour.</i>  |
| PIB  | <i>Policy Information Base.</i>  |
| PID  | Proporcional, Integral e Derivativo.   |
| PoP  | <i>Point of Presence.</i> Ponto de Acesso.   |
| POP3 | <i>Post Office Protocol</i> versão 3.  |
| QBSS | <i>QBone Scavenger Service.</i>  |
| QoS  | <i>Quality of Service.</i> Qualidade de Serviço.   |
| RBAC | <i>Role-Based Access Control.</i> Controlo de acesso baseado em papéis.  |
| RDIS | Rede Digital com Integração de Serviços.   |
| RED  | <i>Random Early Detection.</i> Detecção Antecipada Aleatória.  |
| RFC  | <i>Request For Comment.</i> Uma série de documentos descrevendo o conjunto de protocolos Internet e experiências relacionadas.                           |

|        |   |
|--------|---|
| RIO    | RED com bit <i>in/out</i> .   |
| RM-ODP | <i>Reference Model for Open Distributed Processing</i> .  |
| RMON   | <i>Remote Network Monitoring</i> .  |
| ROSE   | <i>Remote Operations Service Element</i> .  |
| RSA    | Rivest, Shamir, Adleman. Método de encriptação com chaves públicas de [RSA 78].   |
| RSVP   | <i>Resource ReSerVation Protocol</i> .  |
| RTP    | <i>Real-Time Protocol</i> .   |
| SAP/R3 | <i>Systeme, Anwendungen, Produkte in der datenverarbeitung, Release 3</i> . Terceira versão de software de Sistemas, Aplicações e Produtos em processamento de dados.   |
| SHA    | <i>Secure Hash Algorithm</i> .  |
| SLA    | <i>Service Level Agreement</i> . Acordo de níveis de serviço. Definição explícita dos níveis de serviço que se pretende fornecer às várias categorias de utilizadores numa empresa.                                       |
| SLM    | <i>Service Level Management</i> . Gestão de níveis de serviço. A estratégia para definir, controlar e manter níveis requeridos de serviço de tecnologias de informação aos utilizadores de uma empresa.                   |
| SLO    | <i>Service Level Objectives</i> . Implementação dos níveis de serviço acordados num SLA; factor principal para determinar quais os parâmetros a recolher, medir, armazenar, e relatar para ter sucesso em cumprir um SLA. |
| SLR    | <i>Service Level Report</i> . Relatório de níveis de serviço.   |
| SLS    | <i>Service Level Specification</i> . Especificação de níveis de serviço.  |
| SMF    | <i>Systems Management Functions</i> . Funções de Gestão de Sistemas.  |
| SMI    | <i>Structure of Management Information</i> . As regras que definem os objectos que podem ser acedidos através de um protocolo de gestão de redes.   |
| SMSL   | <i>Systems Management Scripting Language</i> .  |
| SMTP   | <i>Simple Mail Transfer Protocol</i> .  |
| SNMP   | <i>Simple Network Management Protocol</i> . O protocolo de gestão de redes normalizado na Internet que permite a comunicação entre agentes e gestores.  |
| SNMPv2 | SNMP versão 2. A segunda versão do SNMP.  |
| SNMPv3 | SNMP versão 3. A terceira versão do SNMP.   |
| SPE    | <i>Software Performance Engineering</i> .   |

|          |   |
|----------|---|
| SQL      | <i>Structured Query Language.</i>   |
| SSH      | <i>Secure Shell.</i>  |
| TCA      | <i>Traffic-Conditioning Agreement.</i> Acordo de condicionamento de tráfego.  |
| Tcl      | <i>Tool Command Language</i> (pronunciado “tickle”). A linguagem de scripts de [Ousterhout 94].   |
| TCP      | <i>Transmission Control Protocol.</i> O protocolo de transporte com ligação de dados utilizado na Internet.                                 |
| TINA     | <i>Telecommunications Information Networking Architecture.</i>  |
| TM Forum | <i>TeleManagement Forum.</i>  |
| TMN      | <i>Telecommunications Management Network.</i>   |
| TOS      | <i>Type Of Service.</i>   |
| UDP      | <i>User Datagram Protocol.</i> O protocolo de transporte sem ligação de dados utilizado na Internet.  |
| UWMA     | <i>Uniformly Weighted Moving Average.</i> Média móvel de peso constante.  |
| VINT     | Virtual InterNetwork Testbed.   |
| VPN      | <i>Virtual Private Network.</i> Rede privada virtual.   |
| WBEM     | <i>Web-Based Enterprise Management.</i>   |
| WDM      | <i>Wavelength-Division Multiplexing.</i> Multiplexagem por divisão no comprimento de onda.  |
| WWW      | <i>World-Wide Web.</i> Um conjunto de servidores com páginas hipertexto que apontam para outras páginas, formando uma teia a nível mundial. |
| XML      | <i>Extensible Markup Language.</i>  |





# APÊNDICE B

## INTRODUÇÃO AOS

### CONTROLADORES PID

A topologia de um sistema genérico de controlo com realimentação está ilustrada na figura B.1 [Franklin 87a], onde  $G$  é o sistema a controlar,  $D$  é o controlador,  $y$  é o sinal de saída que se pretende que siga o sinal de referência  $r$ ,  $e$  é o erro de seguimento, e  $u$  é o sinal de controlo.

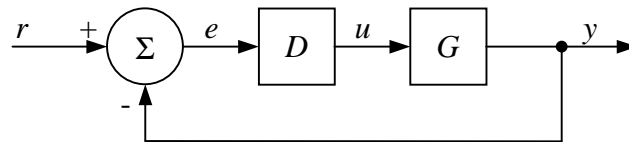


Figura B.1: Topologia de um sistema de controlo.

Há vários tipos de controlador. De entre eles, o controlador **PID** destaca-se pela simplicidade, eficiência e, portanto, generalizada utilização. O nome do controlador é uma sigla que indica a soma ponderada de componentes proporcional, integral e derivativa de erro. Estas componentes são representadas no caso de tempo discreto por:

$$u(n) = K_p e(n) \quad (\text{B.1})$$

$$u(n) = u(n-1) + K_p \frac{T_0}{T_I} e(n) \quad (\text{B.2})$$

$$u(n) = K_p \frac{T_D}{T_0} [e(n) - e(n-1)] \quad (\text{B.3})$$

onde o valor entre parêntesis curvos representa o instante de amostragem, e  $T_0$  o período de amostragem.

Os efeitos principais das componentes proporcional e integral são, respectivamente, tornar a resposta mais rápida, e reduzir o erro estático entre a saída e a referência. Ambas têm como desvantagem um aumento da instabilidade do sistema.

A componente derivativa é normalmente utilizada em conjunto com a proporcional e/ou integrativa, e tem como efeito principal reduzir a sobrelevação da resposta, pois aumenta o atrito do sistema, tornando a resposta mais suave e estabilizando o sistema.

A frequência de amostragem deve ser, de acordo com o teorema da amostragem, superior ao dobro da maior frequência com significado no sistema, também chamada largura de banda do sistema. Esta frequência deve cobrir as frequências dos sinais de entrada, as frequências dos sinais de saída, e ainda as frequências das perturbações que se pretende que o controlador rejeite. Utilizar uma frequência de amostragem superior normalmente origina custos, tanto devido à quantidade de informação que é necessário recolher do sistema, como por reduzir-se o período de tempo disponível para efectuar os cálculos necessários por cada amostra. Por outro lado, uma frequência de amostragem inferior introduz um atraso superior entre o sinal de controlo e a resposta do sistema, o que é indesejável. Na prática, é típico utilizar frequências de amostragem de 10 a 20 vezes superiores à largura de banda do sistema.

A representação em frequência das três componentes de controlo obtém-se através da transformada Z:

$$D(z) = K_p \quad (\text{B.4})$$

$$D(z) = K_p \frac{T_0}{T_I} \frac{1}{1-z^{-1}} \quad (\text{B.5})$$

$$D(z) = K_p \frac{T_D}{T_0} (1-z^{-1}) \quad (\text{B.6})$$

Juntando os três tipos de controlador, obtém-se a função de transferência do controlador PID:

$$D(z) = K_p + K_p \frac{T_0}{T_I} \frac{1}{1-z^{-1}} + K_p \frac{T_D}{T_0} (1-z^{-1}) \quad (\text{B.7})$$

Aplicando a transformada Z inversa, volta-se ao domínio do tempo, obtendo-se a equação às diferenças que permite calcular o sinal de controlo em cada instante de amostragem para o controlador PID:

$$u(n) = u(n-1) + K_p \left[ (e(n) - e(n-1)) + \frac{T_0}{T_I} e(n) + \frac{T_D}{T_0} (e(n) - 2e(n-1) + e(n-2)) \right]$$

(B.8)



# APÊNDICE C

## ANÁLISE ESTATÍSTICA DE RESULTADOS DE SIMULAÇÃO

O objectivo das simulações é obter estimativas fiáveis de medidas de determinados parâmetros de sistemas com custo razoável. Uma análise estatística dos resultados das simulações permite avaliar o grau de confiança nos resultados e saber se é necessário, ou não, prolongar as simulações.

De entre os vários métodos existentes para obter várias amostras de dados, o mais simples e geral é o método das réplicas independentes [Pritsker 86][Mitrani 82]. Neste método, as simulações são realizadas várias vezes, de forma independente, com as mesmas condições iniciais, mas com diferentes sequências de números aleatórios. A utilização deste método é obrigatória quando se pretende estudar o desempenho transitório do sistema. Quando se pretende estudar o comportamento do sistema no equilíbrio, este é um dos métodos possíveis.

Há vários métodos que permitem reduzir a variância dos resultados e assim evitar fazer simulações mais longas. No caso de se pretender estudar o comportamento no equilíbrio, se houver um transitório inicial, pode-se descartar os dados correspondentes ao transitório inicial. Há várias formas de determinar quando o transitório acaba e se está próximo do equilíbrio. Uma forma é [Mitrani 82] considerar que o transitório acaba quando a estimativa do parâmetro medido pára de variar na direcção inicial e começa a oscilar. Outro método de reduzir a variância, quando se pretende comparar várias opções, é utilizar a mesma sequência de números aleatórios em simulações com cada uma das opções, de forma a que as condições sejam o mais idênticas possível para cada uma das opções. Finalmente, outro método de reduzir a variância é substituir variáveis aleatórias pelo seu valor esperado.

O método das réplicas independentes funciona da seguinte forma. A variável aleatória  $X$  representa o parâmetro de interesse do sistema. São feitas  $I$  simulações. Cada simulação  $i$  dá uma amostra  $X_i$  da variável aleatória  $X$ .

A média dos  $X_i$  nas  $I$  simulações é utilizada como uma estimativa do parâmetro de interesse, isto é:

$$\bar{X}_I = \frac{\sum_{i=1}^I X_i}{I} \quad (\text{C.1})$$

Uma estimativa da variância associada a uma amostra aleatória obtida de uma simulação, isto é  $\text{Var}[X_i]$ ,  $S_x^2$ , é obtida pelos procedimentos normais como:

$$S_x^2 = \frac{1}{I-1} \sum_{i=1}^I (X_i - \bar{X}_I)^2 = \frac{1}{I-1} \sum_{i=1}^I X_i^2 - \frac{1}{I(I-1)} \left( \sum_{i=1}^I X_i \right)^2 \quad (\text{C.2})$$

Sendo cada simulação uma replicação independente, uma estimativa da variância da média das amostras,  $S_{\bar{X}}^2$ , é dada por:

$$S_{\bar{X}}^2 = \frac{S_x^2}{I} \quad (\text{C.3})$$

Falta agora determinar o número de simulações necessárias.

Caso se assuma que  $\bar{X}_I$  não é enviesado, e que  $\text{Var}[\bar{X}_I] = \frac{\sigma_x^2}{I}$ , então o número de simulações necessárias, para obter uma confiança de  $(1-\alpha)$  que a média  $\mu_x$  está contida no intervalo pretendido, pode ser calculado utilizando fórmulas estatísticas. Simbolicamente, supondo que se pretende que:

$$P[\bar{X}_I - g \leq \mu_x \leq \bar{X}_I + g] \geq 1 - \alpha \quad (\text{C.4})$$

onde  $g$  é metade da amplitude pretendida para o intervalo de confiança. Fazendo:

$$Z = \sqrt{I}(\bar{X}_I - \mu_x) / \sigma_x \quad (\text{C.5})$$

obtém-se:

$$P\left[|Z| \leq \frac{g\sqrt{I}}{\sigma_x}\right] \geq 1 - \alpha \quad (\text{C.6})$$

sendo a igualdade correspondente ao menor valor de  $I$ , chamêmos-lhe  $I^*$ . Assumindo que o número de simulações,  $I^*$ , é suficientemente grande para que se possa aplicar o

teorema do limite central, isto é, que a acumulação de amostras envolveu a adição de um grande número de pequenas contribuições, pelo que as amostras têm distribuição normal, então:

$$I^* = \left( \frac{\sigma_x}{g} Z_{\alpha/2} \right)^2 \quad (C.7)$$

onde  $Z_{\alpha/2}$  é tal que:

$$\frac{1}{\sqrt{2\pi}} \int_{Z_{\alpha/2}}^{\infty} e^{-y^2/2} dy = \alpha/2 \quad (C.8)$$

visto que  $Z$  converge para uma distribuição normal de média 0 e variância 1, isto é,  $N(0,1)$ .

A equação C.7 para  $I^*$  necessita do conhecimento de  $\sigma_x$ . Um truque comum é especificar  $g$  em termos relativos de  $\sigma_x$ , isto é, fazer:

$$g = v \cdot \sigma_x, \text{ com } v > 0 \quad (C.9)$$

Neste caso,  $I^*$  pode ser calculado sem conhecimento de  $\sigma_x$ . Alguns valores de  $I^*$  para combinações de  $v$  e  $\alpha$  são dados na tabela C.1. Por exemplo, serão necessárias 384 simulações para obter um intervalo de confiança de 95% de que  $\mu_x$  está em  $[\bar{X}_I - 0.1\sigma_x, \bar{X}_I + 0.1\sigma_x]$ .

| $v$         | $I^*$         |               |               |
|-------------|---------------|---------------|---------------|
|             | $\alpha=0.02$ | $\alpha=0.05$ | $\alpha=0.10$ |
| <b>0.01</b> | 54093         | 38416         | 27060         |
| <b>0.10</b> | 541           | 384           | 271           |
| <b>0.20</b> | 135           | 96            | 68            |
| <b>0.50</b> | 22            | 15            | 11            |

Tabela C.1: Valores de  $I^*$  para algumas combinações de  $v$  e  $\alpha$ .

Tipicamente  $S_x$  é utilizado em vez de  $\sigma_x$ , caso em que  $\sqrt{I}(\bar{X}_I - \mu_x)/S_x$  tem uma distribuição de *t-student*. Em estudos de simulação, o número de simulações,  $I$ , é usualmente suficientemente elevado para assumir que a aproximação normal à distribuição de *t-student* é válida.

Nestas condições, [Pritsker 86] sugere a utilização de um número de simulações dado por:

$$I^* = \min \left\{ I : I \geq 3; \quad S_X^2 \leq \frac{I \cdot g^2}{t_{\alpha/2, I-1}^2} \right\} \quad (\text{C.10})$$

onde  $t_{\alpha/2, I-1}$  corresponde ao quantil  $(1-\alpha/2)$  da distribuição *t-student* com  $I-1$  graus de liberdade, por exemplo  $t_{0.05, 9} = 1.833$ .

Assim, para um dado grau de confiança e uma dada amplitude do intervalo de confiança, far-se-iam simulações até verificar a equação C.10.

Note-se que o valor de metade da amplitude do intervalo de confiança é dado por:

$$g = \frac{S_X}{\sqrt{I}} t_{\alpha/2, I-1} \quad (\text{C.11})$$

que será tanto menor quanto menor for a variância, ou quanto maior for o número de simulações.

Como exemplo de aplicação, se tivermos 10 simulações com  $\bar{X}_{10} = 1.982$ , calculado pela equação C.1, e  $S_X^2 = 3.172$ , calculado pela equação C.2, o intervalo de confiança a 90% será:  $\bar{X}_{10} \pm \sqrt{\frac{S_X^2}{10}} t_{0.05, 9} = 1.982 \pm 1.032$ . Isto é, pode-se afirmar, com uma confiança aproximada de 90%, que o valor do parâmetro se encontra no intervalo [0.950, 3.014].

### Resumo da notação utilizada neste apêndice:

|                 |   |
|-----------------|---|
| $X$             | variável aleatória  |
| $X_i$           | amostra da variável aleatória obtida na simulação $i$                             |
| $\bar{X}_I$     | estimativa do valor médio da variável aleatória, isto é, estimativa de $\mu_x$    |
| $\mu_x$         | valor médio da variável aleatória   |
| $I$             | número de simulações realizadas   |
| $I^*$           | menor número de simulações a realizar   |
| $S_X^2$         | estimativa da variância da variável aleatória, isto é, estimativa de $\sigma_x^2$ |
| $S_{\bar{X}}^2$ | estimativa da variância da média das amostras                                     |
| $\sigma_x$      | desvio padrão da variável aleatória   |
| $(1-\alpha)$    | grau de confiança   |



- $g$  metade da amplitude pretendida para o intervalo de confiança
- $v$  metade da amplitude pretendida para o intervalo de confiança, expresso em termos do desvio padrão da variável aleatória
- $t_{\alpha/2, I-1}$  quantil  $(1-\alpha/2)$  da distribuição *t-student* com  $I-1$  graus de liberdade