UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

UMA ABORDAGEM PARA ADAPTAÇÃO DE QoS BASEADA EM CONTROLE NEBULOSO

Tese submetida à
Universidade Federal de Santa Catarina
como parte dos requisitos para a
obtenção do grau de Doutor em Engenharia Elétrica.

CRISTIAN KOLIVER

Florianópolis, Maio de 2001.

"Uma Abordagem para Adaptação de QoS Baseada em Controle Nebuloso"

Cristian Koliver

'Esta Tese foi julgada adequada para obtenção do Título de Doutor em Engenharia Elétrica, Área de Concentração em Sistemas de Informação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Santa Catarina.'

Prof. Jean-Marie Farines, Dr.
Orientador

Prof. Edson Roberto De Pieri, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. Jean-Marie Farines, Dr.
Presidente

Prof. Joni da Silva Fraga, Dr.
Co-orientador

Prof. Otto Carlos Muniz Bandeira Duarte, Dr.

Prof. Klara Nahrstedt, Dra.

Prof. José Marcos Silva Nogueira, Dr.

Prof. Sandra Sandri, Dra.

AGRADECIMENTOS

Agradeço ao todos meus amigos, em especial a Álvaro Freitas Moreira, Heitor Strogulski, Ricardo "Difa" Difenthaeler e Ricardo "Kdinho" Vargas Dorneles, pelos momentos de descontração e pelas discussões "políticas" via correio eletrônico; a turma do Kernel (Alexandre "Sobral" Moreira de Moraes, Augusto "Baiano" César Pinto Loureiro da Costa, Carlos Alberto "Zé Brandão" Barbosa Leite e Lau Cheuk Lung) pelo companheirismo; a César C. Torrico, pela amizade e ajuda no uso do Latex.

A Samira Zurba, pela compreensão e carinho.

À professora Klara Nahrstedt, pelas sugestões e críticas.

Aos professores Jean-Marie Farines e Joni da Silva Fraga, pela orientação e paciência.

Aos meus pais, Enio e Isete... ...

RESUMO

Durante a execução de uma aplicação multimídia distribuída, os parâmetros de qualidade de serviço (QoS) podem sofrer variações bruscas e descontroladas em seus valores em virtude de perturbações externas ao sistema multimídia distribuído. O usuário percebe essa variação na forma de lapsos no som, distorção e estagnação do vídeo e falta de sincronismo entre imagem e som. O objetivo desta tese é propor um mecanismo para adaptação de QoS baseado em controle nebuloso que minimize os efeitos dessas perturbações, aumentando a satisfação do usuário. O uso de controle nebuloso deve-se ao fato desse tipo de abordagem ter sido pouco explorada por mecanismos de adaptação de QoS, apesar de ser uma abordagem que tem sido usada com sucesso para o controle de sistemas com características similares àquelas de um sistema multimídia distribuído. O mecanismo proposto representa a qualidade através combinações de valores de parâmetros de QoS da aplicação com uma métrica associada e obtida a partir da opinião dos usuários. Os resultados obtidos mostraram a adequação do uso de controle nebuloso, no sentido de proporcionar uma adaptação mais ajustada à dinâmica do sistema multimídia distribuído, e da representação de qualidade utilizada, que permitiu um melhor uso da largura de banda disponível, sob o ponto de vista do usuário final.

ABSTRACT

During the execution of a distributed multimedia application, the parameters of quality of service (QoS) can suffer sudden and uncontrolled variations in their values due to perturbations external to the distributed multimedia system (DMS). The end user perceives this variation as lapses in the sound, distortion and stagnation of the image and lack of synchronism between image and sound. The goal of this thesis is to propose a QoS adaptation mechanism based on fuzzy control that minimizes the effects of the perturbations and improves the users' satisfaction. The use of fuzzy control is due to the fact that this approach has not been fully explored for QoS adaptation, despite it has been used successfully to control systems with features close to DMS features. The mechanism represents the quality by combinations of QoS parameters values of the application. A function associates a metric, obtained from users' opinions, to these combinations. The results obtained have showed that fuzzy control is an approach more suitable to the SMD dynamic than classical control. They also showed the quality representation used by the QoS adaptation mechanism permits a better utilization of the network bandwidth.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Acrônimos

ADPCM Modulação Diferencial Adaptativa por Código de Pulso ("Adaptive Differential Pulse Code Modulation")

ATM Modo de Transferência Assíncrono ("Asynchronous Transfer Mode")

bps Bits por segundo ("bits per second")

CN Controlador nebuloso

COG Centro de Gravidade ("Center of Gravity")

CIF Formato Intermediário Comum ("Common Intermediate Format")

DAA Algoritmo de Ajuste Dinâmico ("Dynamic Adjustment Algorithm")

DCT Transformada Discreta de Cosseno ("Discrete Cosine Transform")

DCSQS Escala de Qualidade Contínua de Duplo Estímulo ("Double-Stimulus Continuous Quality Scale")

DPCM Modulação Diferencial por Código de Pulso ("Differential Pulse Code Modulation")

EACM Mecanismo de Controle de Aplicação Fim-a-fim ("End-to-end Application Control Mechanism")

ER Taxa de bits explícita ("explicit rate")

ETSI Instituto Europeu de Padrões de Telecomunicações ("European Telecommunications Standards Institute")

fps Quadros por segundo ("frames per second")

GOP Grupo de quadros ("Group of Pictures")

HDTV Televisão de alta definição ("High Definition Television")

HTML Linguagem para criação de hiperdocumentos ("HyperText Markup Language")

IDCT DCT inversa

IEC Comissão Eletrotécnica Internacional ("International Electrotechnical Commission")

ISO Organização Internacional para Padronização ("International Organization for Standardization")

ITS Intituto para Ciências de Telecomunicação ("Institute for Telecommunication Sciences")

ITU União Internacional de Telecomunicação ("International Telecommunication Union")

ITU-R União Internacional de Telecomunicação - Setor de Radiocomunicação

ITU-TS União Internacional de Telecomunicação - Setor de Telecomunicação

JPEG Grupo de Especialistas em Fotografia ("Joint Photographic Expert Group")

MIDI Interface Digital para Instrumentos Musicais ("Musical Instrument Digital Interface")

M-JPEG JPEG com movimento

MOS Escore de Opinião Média ("Mean Opinion Score")

MPEG Grupo de Especialistas em Fotografias com Movimento ("Motion Photographic Expert Group")

NTSC Comitê Nacional de Sistemas de Televisão ("National Television Systems Committee")

PAL Linha de fase alternativa ("Phase Alternative Line")

QCIF Um quarto de CIF

QoS Qualidade de serviço ("Quality of service")

RSVP Protocolo de Reserva de Recursos ("ReSerVation Protocol")

RTP Protocolo de Transporte de Tempo Real ("Real-Time Transport Protocol")

RM (células) Gerente de Recursos ("Resource Manager")

RR Informe do receptor ("receiver report")

RTCP Protocolo de Controle de Tempo Real ("Real-Time Control Protocol")

SMD Sistema Multimídia Distribuído

SR Informe do emissor ("sender report")

TCP Protocolo de Controle de Transporte ("Transport Control Protocol")

UDP Protocolo de Datagrama do Usuário ("User Datagram Protocol")

VCR Gravador de videocassete ("Videocassette Recorder")

VHS Sistema Doméstico de Vídeo ("Video Home System")

WAN Rede de Longa Distância ("Wide Area Network")

Capítulo 1

INTRODUÇÃO

1.1 Conceitos Básicos

Conforme F. Fluckiger (Fluckiger 1995), uma *mídia contínua* ou *dependente do tempo* é um tipo de mídia no qual o tempo é parte da informação. Alguns exemplos de mídias contínuas são: gráficos em movimento (como animações e simulações), fotos em movimento (como filmes e clipes) e som (música, fala, comandos MIDI[1], ...). De maneira oposta, uma *mídia discreta* é uma mídia na qual não existe uma dependência temporal na apresentação da informação que possa causar a perda do significado original. São exemplos de mídias discretas: textos não-formatados e formatados (como "slides", "scripts" e livros), textos estruturados (como HTML), gráficos (desenhos, ...) e imagens estáticas. Fluckiger salienta, contudo, que a linha que separa esses dois conjuntos é tênue, já que geralmente apresentações de textos e gráficos baseadas em computador têm algum tipo de dependência temporal.

No contexto da informática, uma *aplicação multimídia* é uma aplicação que integra várias mídias, incluindo tanto mídias contínuas quanto discretas. Aplicações multimídia concebidas para serem executadas em um ambiente distribuído aberto são referenciadas como *aplicações multimídia distribuídas* e o conjunto de entidades que permite a execução dessas aplicações é designado como *Sistema Multimídia Distribuído* (SMD).

A maior parte das aplicações de informática possui um ou mais requisitos em relação ao ambiente de execução: necessidades de armazenamento, processamento, memória RAM etc. Também os usuários dessas aplicações têm suas exigências ou requisitos em relação a aspectos como grau de tolerância a falhas, precisão dos dados e tempo de resposta. Tais requisitos têm si-

[1] "Musical instrument digital interface".

do referenciados, nos últimos anos, como *requisitos de qualidade de serviço* (ou simplesmente, requisitos de QoS) e os atributos do sistema que são alvo dessas exigências são referenciadas como *parâmetros de QoS*. Em aplicações multimídia distribuídas, devem ser acrescidos à gama de parâmetros de QoS anteriormente citados aqueles parâmetros ligados diretamente à qualidade de exibição ou reprodução dos dados das mídias contínuas, como resolução e freqüência de quadros em uma animação e freqüência de amostras de áudio para o som, e outros cuja influência nessa qualidade é indireta, como atraso e variação do atraso fim-a-fim, taxa de bits, taxa de perdas de pacotes e taxa de perdas de "deadlines".

1.2 Motivação

Todos esses parâmetros de QoS que influenciam na qualidade percebida pelo usuário final podem sofrer variações bruscas e descontroladas em seus valores durante as transferências dos dados na rede ou durante sua apresentação em virtude da variação da carga da rede e dos processadores dos sistemas finais. O usuário percebe essa variação na forma de lapsos no som, distorção e estagnação da imagem e falta de sincronismo entre imagem e som.

Com a popularização das aplicações multimídia, têm crescido muito o número de pesquisas que tem como alvo a definição e construção de modelos que, dentro de SMD's, realizem a gerência da QoS, permitindo que a qualidade final percebida pelo usuário mantenha-se estável ou até sofra uma degradação, mas de forma suave e dentro de suas expectativas, isto é, dentro de certos limites e de acordo com suas preferências quando do congestionamento da rede e/ou sobrecarga dos processadores. Esse processo de alteração controlada dos valores dos parâmetros de QoS, face a mudanças no contexto corrente do SMD, é referenciado como *adaptação de QoS* (Gecsei 1997). O *mecanismo de adaptação de QoS* é a entidade responsável pelo processo de adaptação. Ele atua sobre a aplicação multimídia distribuída com o intuito de ajustá-la ao contexto corrente do SMD.

Para tratar com a variação da carga do processador, os esforços, em termos de adaptação de QoS, têm sido direcionados para a definição e construção de sistemas operacionais de tempo real que, através de políticas de escalonamento específicas, privilegiem as tarefas relacionadas ao processamento dos dados das aplicações multimídia bem como mecanismos de adaptação que tentem manter os períodos e tempos de processamento de tais tarefas compatíveis com a carga do processador (Chatterjee e Strosnider 1995) (Kawachiya et al. 1995) (Knightly e Zhang 1996)

(Koren e Shasha 1995) (Mercer et al. 1993) (Mercer e Tokuda 1994) (Li e Nahrstedt 1998) (Maruchek e Strosnider 1995) (Nakajima e et al. 1991) (T. Nakajima 1994) (Ramanathan 1997) (Zhang e Knightly 1995) (Waldeegg 1997). Outras soluções envolvem o uso de "hardware" específico para multimídia (codificadores, decodificadores e placas de som, cujo custo tem caído de forma crescente).

Para tratar com a variação da carga da rede - o escopo deste trabalho -, as abordagens têm sido direcionadas para a definição e construção de mecanismos de adaptação que mantenham a taxa de bits das aplicações multimídia em um nível compatível com a disponibilidade de largura de banda (Baiceanu et al. 1996) (Bocheck et al. 1999) (Bolot et al. 1994) (Bolot e Turletti 1998) (Busse et al. 1995) (Campbell et al. 1998) (N.G. Duffield 1998) (Eleftheriadis e Anastassiou 1995) (Fry et al. 1996) (Fukuda et al. 1998) (Gonçalves et al. 2000) (Koliver et al. 2000 a) (Lakshman et al. 1997) (Li et al. 1998) (McCanne et al. 1996) (Ott et al. 1996) (Silveira e Ruggiero 2000) (Sisalem e Schulzrinne 2000) (Welling et al. 1996) (Yeadon et al. 1996) ou realizem um balanceamento da carga da rede (Nahrstedt e Steinmetz 1995) (Fischer, Salem e Bochmann 1997), através de um roteamento dinâmico. Uma outra abordagem baseia-se na reserva de recursos na rede. Algumas alternativas neste sentido fazem uso de tecnologias de rede específicas (como ATM) ou de protocolos que permitem implementar serviços integrados ou diferenciados (Braden et al. 1994) (Nichols et al. 1998) (Zhang et al. 1993). A reserva de recursos, apesar de ser condição *sine qua non* para tornar as redes, de fato, competitivas com os meios tradicionais de distribuição em tempo real de áudio e vídeo (telefone, rádio e televisão), só elimina a necessidade da adaptação de QoS se ela for realizada considerando-se sempre o pior caso, isto é, a taxa de bits máxima gerada pela aplicação. Em virtude do uso de algoritmos de compressão, tal taxa pode ser diversas vezes maior do que a taxa média, chegando à ordem de vários megabits, o que ocasiona uma sub-utilização da rede. Quando a reserva não é feita considerando o pior caso, surge novamente a necessidade de adaptação de QoS. Logo, a reserva de recursos e a adaptação de QoS não são abordagens mutuamente exclusivas.

De maneira genérica, um mecanismo de adaptação de QoS deve fazer uso de uma política que defina as situações nas quais o processo de adaptação será disparado, os parâmetros para os quais haverá um esforço maior na preservação da QoS, os que serão prioritariamente degradados, os valores que eles poderão assumir etc. No caso específico da adaptação à carga da rede, o mecanismo deve especificar que ações de adaptação serão, de fato, realizadas, isto é, como a aplicação terá sua taxa de bits alterada em consonância com a política usada.

Contudo, a maior parte dos trabalhos propostos na literatura relacionados a mecanismos de adaptação de QoS à carga da rede apresenta um ou mais dos seguintes problemas:

1. as políticas são limitadas, restringindo-se à definição de valores ou intervalos de uma variável observada (por exemplo, freqüência de quadros, atraso ou taxa de perdas de pacotes) com ações associadas (por exemplo, reduzir a taxa de bits à $x\%$) sem especificar como tais ações serão, de fato, realizadas;

2. as ações de adaptação atuam, em geral, apenas sobre um único parâmetro de QoS, desconsiderando que o usuário final percebe a qualidade como um todo, isto é, de forma multidimensional;

3. as políticas não contemplam a incerteza inerente à determinação do estado de um SMD, um tipo de sistema extremamente dinâmico e sujeito a perturbações cujo comportamento é imprevisível;

4. muitas políticas privilegiam a melhor utilização dos recursos em detrimento do usuário final; e

5. as abordagens utilizadas são, em geral, amarradas a tecnologias de rede ou algoritmos de compressão específicos.

1.3 Objetivo

O objetivo deste trabalho é a definição e verificação da viabilidade de um mecanismo de adaptação de QoS que possibilite que a qualidade das aplicações multimídia mantenha-se dentro das expectativas do usuário final mesmo quando da ocorrência de flutuações na carga da rede.

Para tratar com o aspecto da incerteza do estado do SMD mencionado acima, o mecanismo de adaptação de QoS proposto faz uso de Lógica Nebulosa. Três características dos SMD's - não linearidade no relacionamento entre os parâmetros de QoS, imprecisão na determinação do estado do sistema e o critério de desempenho (qualidade) vago - justificam o uso de um modelo de representação e de controle baseado em lógica nebulosa.

Ao contrário da maior parte das abordagens existentes, o mecanismo aqui proposto age sobre o maior número possível de parâmetros de QoS, de modo que cada um possa contribuir um pouco para manter a qualidade dentro de um patamar pré-estabelecido, evitando uma degradação demasiadamente grande sobre um único parâmetro.

Duas das principais características do mecanismo de adaptação de QoS proposto são sua *generalidade* e *orientação ao usuário final.* A primeira característica é obtida porque o mecanismo: (1) não assume qualquer premissa a respeito de plataformas (tecnologia de rede, protocolos de comunicação, sistema operacional etc.); (2) não assume qualquer premissa a respeito de algoritmos da compressão e parâmetros de QoS a serem adaptados; e (3) pode ser usado tanto em ambientes que oferecem algum tipo de garantia de QoS quanto em ambientes totalmente melhor-esforço. A segunda característica é obtida porque o mecanismo é baseado na definição da *função do grau da qualidade* que conduz todo o processo da adaptação. Tal função associa uma medida arbitrária do desempenho às diversas combinações de valores de parâmetros de QoS e é construída de acordo com as preferências dos usuários obtidas a partir de avaliações empíricas. Ela é usada pelo mecanismo de adaptação de QoS para realizar o mapeamento entre a qualidade, de acordo com o ponto de vista do usuário final, e valores de parâmetros de QoS da aplicação. O uso da função grau da qualidade permite que o mecanismo de adaptação de QoS maximize a qualidade como um todo ao invés de otimizar um único parâmetro de QoS, como em outra propostas.

1.4 Organização do Texto

O presente trabalho é assim constituído:

- no Capítulo 2 são apresentadas algumas das definições para QoS comumente encontradas na literatura, a visão arquitetural de um SMD que será usada no decorrer do trabalho e algumas taxinomias propostas para os parâmetros de QoS. Neste capítulo são também discutidos vários aspectos relacionados à qualidade (formas de avaliação, fatores que influenciam e relação entre qualidade e largura de banda);

- no Capítulo 3 é apresentada a função grau de qualidade, utilizada para fornecer os valores de uma métrica utilizada para avaliar qualidade e peça-chave na abordagem de adaptação proposta neste trabalho;

- no Capítulo 4 são discutidos alguns aspectos relacionados à adaptação de QoS (particularmente, políticas e mecanismos de adaptação);

- no Capítulo 5, são descritos trabalhos relacionados à adaptação de QoS, o mecanismo de adaptação de QoS baseado em controle nebuloso aqui proposto, duas implementações

possíveis e os resultados obtidos a partir dessas implementações;

- no Capítulo 6 é descrita uma proposta de um modelo distribuído para adaptação de QoS; e
- no Capítulo 7 são apresentados um resumo do trabalho, conclusões e perspectivas para trabalhos futuros.

O trabalho também contém dois apêndices. O Apêndice A fornece uma introdução às técnicas de compressão mais relevantes em aplicações multimídia; o Apêndice B fornece uma introdução à teoria que embasa o controle nebuloso.

Capítulo 2

QUALIDADE DE SERVIÇO

2.1 Introdução

Neste capítulo, são apresentadas algumas das definições de QoS comumente encontradas na literatura, a visão arquitetural de um SMD que será utilizada neste trabalho e o posicionamento dos parâmetros de QoS dentro dessa arquitetura. Neste capítulo, é também proposta uma taxinomia para aplicações multimídia distribuídas voltada para aspectos de QOS e são discutidos vários aspectos que influenciam a qualidade percebida pelo usuário final. A partir dessa discussão, é proposta a divisão dos parâmetros de QoS em categorias. No final do capítulo, é fornecida uma visão geral das formas de avaliação de qualidade propostas na literatura.

2.2 Definição de QoS

Apesar de ser objeto de muitos estudos nos últimos anos, ainda não há uma definição única para qualidade de serviço (QoS). Inicialmente, essa expressão foi introduzida pela "International Standard Organization" (ISO) (ISO 1984) para descrever um conjunto de características relacionadas à transmissão de dados em sistemas de comunicação. Para (Fluckiger 1995), *QoS é o conjunto de requisitos específicos das aplicações sobre a rede na qual elas serão executadas, definidas antes do início da transmissão dos dados.* Conforme esses conceito, a QoS é restrita àqueles requisitos ligados às entidades do sistema de comunicação. Essa visão é encontrada na maior parte dos trabalhos sobre SMD's.

Definições recentes de QoS são mais abrangentes, cobrindo todos os componentes da arquitetura de um SMD, e centradas no ponto de vista do usuário final.

A "European Telecommunications Standards Institute" (ETSI) descreve, no livro "The state of the art 1995", QoS como sendo "*características, avaliáveis qualitativamente ou quantitativamente, que influenciam no desempenho de um serviço*".

No projeto RACE (RAC n.d.), da "European Commission", esse conceito foi estendido: "*QoS é a medida de qualidade do serviço do ponto de vista do usuário. Essa medida é expressada em uma linguagem compreensível pelo usuário e representa vários parâmetros de valores objetivos ou subjetivos*".

Em (Vogel et al. 1995), QoS é definida como sendo um *conjunto de parâmetros, representando características quantitativas e qualitativas de um SMD que são necessárias para a obtenção da funcionalidade desejada de uma aplicação, onde a funcionalidade inclui a apresentação de dados multimídia para o usuário e sua satisfação com essa apresentação.*

Neste trabalho, adotar-se-á uma definição de QoS próxima da visão do usuário, como esta fornecida em (Vogel et al. 1995).

2.3 Sistema Multimídia Distribuído e Parâmetros de QoS

Neste trabalho, é proposta uma visão arquitetural de um SMD consistindo de cinco camadas de abstração refletidas em componentes concretos do sistema: camada do usuário, camada da aplicação, camada do sistema, camada de comunicação e camada dos dispositivos (Koliver et al. 1999). A Figura 2.1 mostra essa arquitetura.

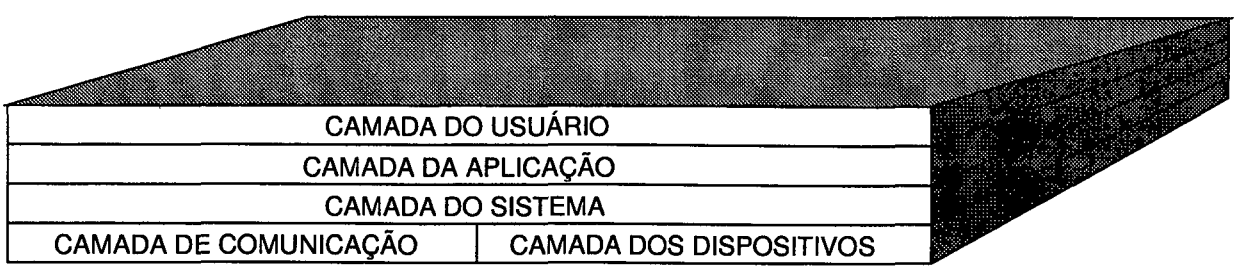

Figura 2.1: Arquitetura de um Sistema Multimídia Distribuído.

Na *camada do usuário* encontram-se as pessoas que irão interagir com o sistema de forma direta ou indireta. Os principais parâmetros dessa camada são:

- qualidade do som;
- qualidade da imagem; e

- tempo de resposta.

A maior parte dos parâmetros da camada do usuário são valorados de forma subjetiva.

A *camada da aplicação* também possui parâmetros de QoS diretamente relacionados à qualidade de apresentação, porém representados por atributos quantificáveis (métricas de QoS) que compõem os parâmetros da camada do usuário. São parâmetros de QoS da camada da aplicação:

- "dessincronização" intermídias ou "skew", em milissegundos (ms);
- freqüência de quadros ou resolução temporal, em quadros por segundo ("frames per second" ou fps);
- tamanho do quadro, em bytes;
- freqüência de amostras de áudio, em amostras por segundo (geralmente expressa em Hz);
- tamanho da amostra de áudio, em bits;
- resolução espacial, em pixels×pixels;
- profundidade do pixel ou resolução cromática, em bits;
- atraso e variação do atraso fim-a-fim, em ms;
- algoritmo de compressão, com os parâmetros relacionados; e
- taxa de compressão.

Além desses parâmetros, estão situados na camada da aplicação vários outros, como aqueles relacionados a aspectos como segurança e tolerância a falhas (grau de confidencialidade, integridade e precisão dos dados).

A *camada do sistema* é composto pelos "middlewares" (com destaque para o sistema operacional) que interagem com a camada da aplicação, camada de comunicação e camada dos dispositivos. Os principais parâmetros de QoS dessa camada relacionam-se aos atributos das tarefas tais como:

- tempo de processamento;
- variação no tempo de processamento;
- "deadline";

- período; e
- prioridade.

Com exceção da prioridade, todos esses parâmetros são expressos em ms.

A *camada de comunicação* é composta por todas as entidades físicas ou abstratas relacionadas ao processo de comunicação (conexões, roteadores, protocolos, unidades de transporte); os parâmetros de QoS dessa camada correspondem aos atributos das camadas de rede, transporte, enlace e física do modelo OSI. Dentre os parâmetros da camada de comunicação, destacam-se:

- largura de banda da rede, em bits por segundo (bps);
- taxa de bits[1], em bps;
- taxa de perdas de pacotes;
- taxa de erros de bits; e
- atraso e variação do atraso ou "jitter" de rede, em ms.

A *camada dos dispositivos* é composta pelos dispositivos de E/S, processadores e memórias dos sistemas finais[2]; os parâmetros de QoS dessa camada correspondem aos atributos desses dispositivos. São parâmetros de QoS da camada de dispositivos:

- poder de processamento, em milhões de instruções de ponto flutuante por segundo (megaflops);
- capacidade de memória, em bytes;
- tempo de busca ("seek"), em ms;
- largura de banda dos dispositivos de armazenamento, em bps;
- latência rotacional, em ms;

[1]Usualmente, as expressões "largura de banda" e "taxa de bits" são utilizadas indistintamente para referenciar a taxa de transmissão gerada por uma aplicação. Neste trabalho, contudo, a expressão "largura de banda" será usada apenas para designar a capacidade de um canal de comunicação da rede.

[2]Na literatura, existem diferentes designações para as entidades que participam da troca de informações: servidor e cliente, fonte e destino, sistemas hospedeiros ("hosts") etc. Neste trabalho, o ponto da rede que inicia o envio do dado será sempre referenciado como *sistema final emissor* e o que recebe o dado como *sistema final receptor*.

- dimensão do monitor, em polegadas; e
- resolução do monitor, em pixels×pixels.

As camadas da arquitetura proposta interagem entre si, através do oferecimento de serviços com determinada QoS. Por exemplo, a camada do sistema oferece para a camada da aplicação acesso a dispositivos de "hardware" da camada de dispositivos. Essa interação reflete-se numa relação de mapeamento entre os parâmetros. Por exemplo, a especificação, na camada da aplicação, de um fluxo com uma freqüência de quadros x com uma resolução espacial y reflete-se, na camada de comunicação, em um fluxo com uma taxa de bits z. Tal relação pode ocorrer também entre parâmetros de QoS não quantificáveis - que representam, em geral, a presença ou não de determinado recurso no sistema ou o uso de determinada política, protocolo ou tecnologia - e métricas de QoS. Por exemplo, a tarefa de descompressão terá um tempo de processamento x se o sistema final dispõe de decodificador[3] implementado via "hardware" ou y ($y > x$) se a descompressão é feita via "software".

Essa arquitetura é semelhante àquelas propostas por K. Nahrstedt e R. Steinmetz em (Nahrstedt e Steinmetz 1995) e A. Hafid, G. von Bochmann e R. Dssouli em (Hafid et al. 1998). Com relação à primeira, a diferença principal é que a camada do sistema nesta abrange os parâmetros dos sistemas de comunicação e operacional (tempo de processamento, taxa de erros de bits, tamanho da unidade de dados, taxa de bits, ...) da arquitetura proposta e a camada de comunicação é referenciada como *camada de rede*, abrangendo parâmetros como atraso e variação do atraso de rede. Essa distinção entre parâmetros de rede e parâmetros do sistema de comunicação não parece tão clara como a divisão entre parâmetros do sistema operacional (e "middlewares") e parâmetros de comunicação aqui proposta. Com relação à segunda, a principal diferença está no número de camadas (nove). Na arquitetura de Hafid et al., o SMD é dividido em componentes, hierarquicamente distribuídos em camadas de maneira similar à da arquitetura proposta no início da seção: no topo da hierarquia está o usuário; a seguir, vem a aplicação multimídia e abaixo, o sistema operacional; na base, em um mesmo nível, estão os sistemas finais, protocolos de transporte, rede, sistema de arquivos multimídia e base de dados multimídia. Uma divisão que contemple camadas da arquitetura exclusivas para o sistema de

[3]Um codificador ("encoder") é uma implementação específica de um algoritmo de compressão, no que diz respeito ao processo de compressão. De maneira análoga, um decodificador é uma implementação de um algoritmo de compressão, no que refere-se à descompressão. Diferentes implementações de codificadores/decodificadores podem exigir mais ou menos recursos, tanto em termos de largura de banda quando poder de processamento, bem como oferecer diferentes funcionalidades.

arquivos e base de dados é interessante apenas em um contexto centrado em aplicações de dados armazenados (como vídeo sob demanda). Dentro dos propósitos deste trabalho, uma arquitetura mais simples, composta de menos camadas, é suficiente.

2.4 Parâmetros de QoS e a Natureza da Aplicação

A *natureza da aplicação* é uma abstração proposta neste trabalho que embute aspectos como o propósito da aplicação multimídia, funcionalidade e público-alvo.

A natureza da aplicação é importante por impor limites nos valores dos parâmetros de QoS, restringindo, assim, a faixa de atuação do mecanismo de adaptação. Por exemplo: atraso fim-a-fim máximo tolerável, resolução e frequência de quadros mínimas etc. A natureza da aplicação também pode limitar os valores alcançáveis para alguns parâmetros de QoS. Por exemplo, a taxa de compressão máxima alcançável, que influi na taxa de bits mínima, guarda relação íntima com o tipo de vídeo que está sendo exibido.

Além de estabelecer limites para determinados parâmetros de QoS, a natureza da aplicação estabelece uma relação de importância entre os parâmetros. Em uma videoconferência do tipo seminário, por exemplo, a preservação da qualidade do som deve ter prioridade sobre a da imagem.

A relação entre a natureza da aplicação e parâmetros de QoS exposta acima implica na necessidade de uma taxinomia de aplicações multimídia direcionada para aspectos de QoS, que pode ser de grande valia quando do projeto de mecanismos de adaptação de QoS.

A "International Telecommunications Union" (ITU) propõe uma taxinomia que divide as aplicações em aplicações de conversação, recuperação de informações, distribuição e mensagens. O ATM fórum propõe outra na qual as aplicações multimídia são divididas em cinco categorias: "broadcast", videoconferência, "desktop", áudio + dados e vídeo sob demanda. Tais classificações, contudo, são muito genéricas por não considerarem as especificidades das aplicações em relação à QoS.

Neste trabalho, é proposta uma taxinomia (Figura 2.2) que divide as aplicações segundo aspectos considerados relevantes em termos de QoS e que devem ser levados em conta quando do projeto de um mecanismo de adaptação de QoS. Os critérios usados para dividir as aplicações foram: manutenção da QoS, interatividade, fonte de obtenção dos dados, conteúdo dos quadros, sincronização entre os fluxos e distribuição dos dados.

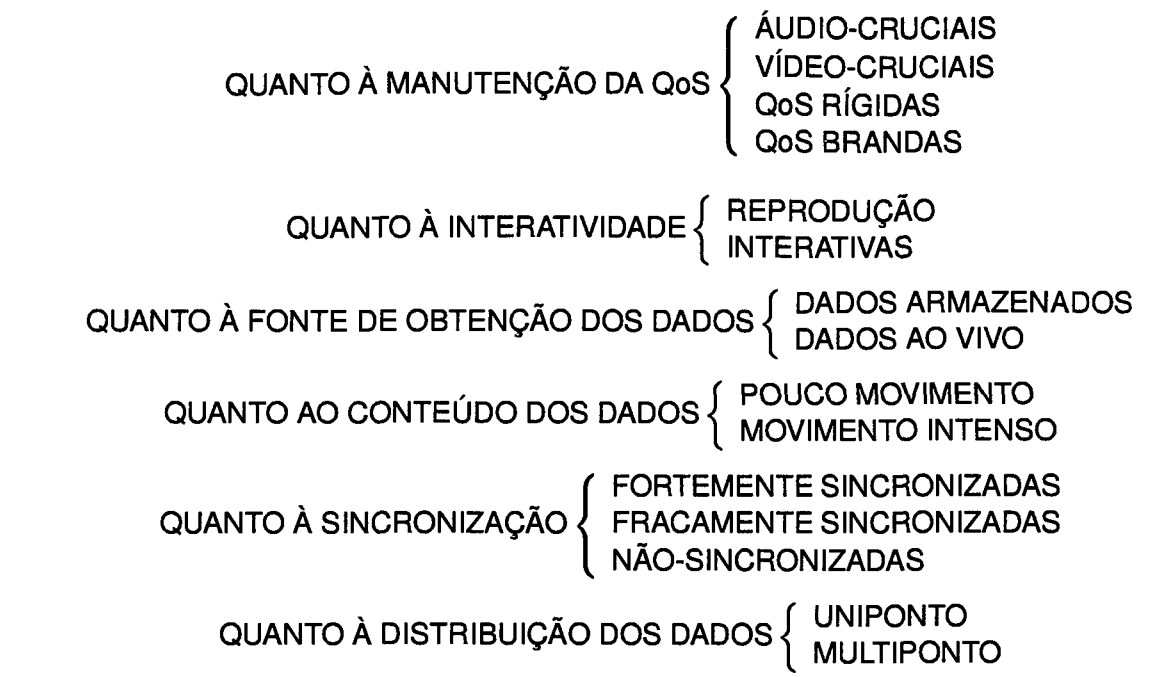

Figura 2.2: Taxinomia das aplicações em relação à QoS.

2.4.1 Manutenção da QoS

Com relação à *manutenção da QoS*, as aplicações podem ser QoS-rígidas, QoS-brandas, vídeo-cruciais ou áudio-cruciais.

Aplicações QoS-rígidas são aplicações onde uma violação de QoS pode ocasionar efeitos catastróficos, em termos de danos materiais ou à vida humana. Exemplos de tais aplicações são telemedicina e controle remoto de sondas espaciais. Aplicações QoS-rígidas necessitam de garantias de serviço determinista[4] e de intervalos de tempo menores para monitoração dos valores dos parâmetros de QoS. O nível de garantia de serviço determinista, por sua vez, implica em reservas de recursos (ciclos de processador e largura de banda de rede, especialmente) para o pior caso. Em termos de plataforma de suporte, é bastante adequado o uso de uma tecnologia de sistema operacional com suporte para tarefas de tempo real "hard" e um meio de comunicação físico (preferencialmente, fibra ótica). Aplicações QoS-rígidas geralmente não são muito flexíveis em termos dos valores de parâmetros de QoS, tolerando uma adaptação dentro de pequenos intervalos.

[4]O tipo de garantia do serviço especifica o grau de comprometimento de recursos fim-a-fim requisitado. O tipo que oferece o grau mínimo é o melhor-esforço ("best-effort"), sem nenhuma garantia; o tipo que oferece o grau máximo é o determinista (garantia de 100%); graus intermediários representam um serviço com alguma previsibilidade, conforme a classificação proposta por R. Nagarajan (Nagarajan 1993).

As *aplicações QoS-brandas*, por outro lado, toleram que a garantia de serviço seja determinista, melhor esforço ou previsível, conforme o que o usuário está disposto a pagar. Assim, a reserva de recursos pode ser baseada no cálculo do pior caso, no do melhor caso ou em estimativas de caso médio. Aplicações QoS-brandas são, intrinsecamente, mais flexíveis em termos dos valores dos parâmetros de QoS, admitindo faixas maiores de atuação do mecanismo de adaptação.

As *aplicações vídeo-cruciais* são aplicações multimídia nas quais o vídeo é mais importante que o áudio. Assim, o mecanismo de adaptação deve tentar preservar a qualidade da imagem em detrimento da qualidade do som. Nas *aplicações áudio-cruciais*, em contrapartida, deve ser realizada uma preservação da qualidade do som em detrimento da qualidade da imagem. Uma aplicação de controle remoto de sondas espaciais é vídeo crucial, enquanto uma aplicação de vídeo-fone é áudio-crucial.

2.4.2 Interatividade

Com relação à *interatividade*, as aplicações podem ser interativas (Homem-Homem) e de reprodução. As *aplicações interativas* são aplicações que não toleram valores altos para o atraso fim-a-fim, variação do atraso e "dessincronização": a variação máxima tolerável do atraso fim-a-fim deve ser inferior a 250 ms e a "dessincronização" máxima tolerável deve ser inferior a 100 ms (Steinmetz 1995). Este requisito, em termos de adaptação, limita a possibilidade de armazenamento temporário dos dados ("buffering") para diluição da variação do atraso fim-a-fim, já que isso aumenta o tempo de resposta. Se o mecanismo de adaptação utilizar ações baseadas na supressão de quadros, ele deve também prever ações para tratar com a "dessincronização" decorrente. Toda aplicação interativa é uma aplicação de dados "ao vivo"[5] (vide adiante), mas a recíproca não é verdadeira.

Aplicações de reprodução são um tipo peculiar de aplicações de dados armazenados (vide seção seguinte) onde são oferecidas operações VCR ("vídeo-cassete recorder"), como retrocesso e avanço normal, lento e rápido, com e sem exibição, e pausa. Tais operações exigem "buffers" com grande capacidade de armazenamento e podem causar mudanças bruscas nos valores de parâmetros de QoS como freqüência de quadros, período e "deadlines" das tarefas.

[5]Dados ao vivo ("live data") são dados que, após adquiridos, são imediatamente transmitidos para o(s) sistema(s) final(is) receptor(es), onde são exibidos; isso ocorre em várias aplicações multimídia: telemedicina, ensino à distância interativo, videoconferência, etc.

O tempo de resposta a esses comandos também deve ser pequeno, similar ao de um aparelho de videocassete (não mais que meio segundo).

2.4.3 Fonte de Obtenção dos Dados

Com relação à *fonte de obtenção dos dados*, as aplicações podem ser aplicações de dados ao vivo ou de dados armazenados.

Nas *aplicações de dados ao vivo*, os atributos das tarefas relacionadas à compressão de quadros (tempo de processamento, "deadline", período) dos sistemas finais emissores são importantes e a possibilidade de sobrecarga dos processadores desses sistemas deve ser considerada.

Nas *aplicações de dados armazenados*, deve-se ter um cuidado maior com parâmetros relacionados a dispositivos de armazenamento secundário (latência de disco, capacidade de armazenamento e largura de banda de disco) e com as políticas de escalonamento de disco. Por outro lado, a não realização de tarefas relacionadas à compressão dos dados torna menos provável a ocorrência de sobrecarga dos processadores desses sistemas finais, mas impossibilita o uso de técnicas de adaptação baseadas na mudança dinâmica de valores dos parâmetros do codificador. Nesse tipo de aplicação, o atraso fim-a-fim não tem a mesma importância do que nas aplicações de dados ao vivo interativas, já que ele é percebido pelo usuário apenas como uma demora inicial na exibição do vídeo.

2.4.4 Conteúdo dos Dados

Com relação ao *conteúdo dos dados*, as aplicações podem ser de movimento intenso ou pouco movimento.

Aplicações de movimento intenso são aplicações onde ocorrem grandes variações no conteúdo dos quadros. Essa variação reduz bastante a taxa de compressão obtida, em virtude da pouca correlação temporal existente entre os quadros que é explorada pelos principais algoritmos de compressão para vídeo. Isso ocasiona a necessidade de taxas de bits mais elevadas. A pouca correlação temporal também faz com que sejam produzidos mais quadros do tipo I[6] (se o algoritmo de compressão usado é, por exemplo, da família MPEG, H261 ou H263), cujas

[6]Um quadro do tipo I é um quadro comprimido que contém todas as informações necessária para sua descompressão; em contrapartida, um quadro do tipo P é um quadro que necessita de informações de um quadro do tipo I para ser descomprimido, enquanto um quadro do tipo B necessita de informações de quadros dos tipos I e P para ser descomprimido. Para maiores detalhes, vide Apêndice A.

tarefas de compressão e descompressão possuem tempos de processamento bastantes superiores àquelas dos quadros dos tipos B e P. Em aplicações de movimento intenso, por outro lado, a "dessincronização" entre o áudio e vídeo não é muito percebida pelo usuário.

Aplicações de pouco movimento são aplicações onde não há muita variação entre o conteúdo dos quadros, como videoconferência do tipo seminário[7].

2.4.5 Sincronização entre os Fluxos

Com relação à *sincronização entre os fluxos*, as aplicações podem ser com fluxos não-sincronizados, com fluxos fracamente sincronizados e com fluxos fortemente sincronizados.

Aplicações com fluxos não-sincronizados são aplicações onde não existem fluxos a serem sincronizados, permitindo que o mecanismo de adaptação não se preocupe com a "dessincronização".

Algumas aplicações possuem diferentes fluxos que devem ser levemente sincronizados (por exemplo, fluxo de áudio e fluxo de caracteres representando legendas ou fluxo de áudio e fluxo de vídeo quando a sincronização labial não é tão importante e a "dessincronização" pode alcançar até 100 ms). Tais aplicações serão referenciadas como *aplicações com fluxos fracamente sincronizados.*

Aplicações onde a "dessincronização" entre os fluxos deve ser inferior a 80 ms serão referenciadas como *aplicações com fluxos fortemente sincronizados.* Este é o caso de aplicações de reprodução de filmes e clipes musicais.

2.4.6 Distribuição dos Dados

Com relação à *distribuição dos dados*, as aplicações podem ser uniponto ("unicast") e multiponto ("multicast"). Nas *aplicações uniponto*, há apenas dois sistemas finais comunicando-se, o que torna bem mais simples o mecanismo de adaptação e possibilita a execução de ações de adaptação no sistema final emissor para aliviar a carga do processador do sistema final receptor. Nas *aplicações multiponto*, todos sistemas finais são emissores e receptores, o que torna mais difícil o controle de QoS: uma ação de adaptação de QoS em um sistema final pode alterar a QoS de outro. Além disso, a adaptação da QoS no ponto de entrada na rede deve ser feita com critério, já que isso afeta todos os sistemas finais receptores da mídia contínua.

[7]Muitas vezes, a classificação em movimento intenso ou pouco movimento é inerente à natureza do vídeo que está sendo exibido e não à aplicação em si.

2.4.7 Enquadramento das Aplicações

As categorias apresentadas anteriormente não são mutuamente exclusivas, *i.e.*, o fato de uma aplicação pertencer a uma determinada categoria não a exclui, necessariamente, de pertencer também à outra: em geral, uma aplicação de vídeoconferência do tipo seminário, por exemplo, é uma aplicação fracamente sincronizada, interativa Homem-Homem, de dados ao vivo, multiponto, de pouco movimento e áudio-crucial; uma aplicação de vídeo sob demanda geralmente é fortemente sincronizada, de movimento intenso e de reprodução de dados armazenados; uma aplicação de telemedicina é uma aplicação não sincronizada (ou pouco sincronizada), de pouco movimento, de dados ao vivo, uniponto e vídeo-crucial. Para certas aplicações muito peculiares, contudo, o enquadramento em algumas categorias é complicado, exigindo futuramente um refinamento da taxinomia conforme a necessidade.

2.5 Fatores que Influenciam a Qualidade

Todos os parâmetros de QoS podem influenciar direta ou indiretamente na qualidade da apresentação de uma mídia contínua. Com o intuito de tornar mais clara essa influência, neste trabalho é proposta uma divisão dos parâmetros de QoS em três categorias: parâmetros de contexto, parâmetros controláveis e parâmetros não-controláveis.

2.5.1 Parâmetros de Contexto

Os *parâmetros de contexto* são parâmetros de QoS relacionados ao ambiente onde a aplicação está sendo executada. Eles têm seus valores definidos ou são instanciados quando da configuração da aplicação multimídia, permanecendo invariáveis durante todo período da sessão da aplicação[8]. A maior parte dos parâmetros de contexto refere-se a dispositivos de "hardware", protocolos de comunicação e "middlewares". Dentre esses parâmetros destacam-se:

- algoritmo de compressão e tipo de codificador/decodificador;
- poder de processamento dos sistemas finais;
- memória do sistema final;

[8]A expressão "sessão da aplicação"(ou, simplesmente, sessão) é usada para designar o período de tempo no qual a aplicação multimídia está ativa, isto é, está sendo executada.

- largura de banda máxima disponível;
- tecnologia de rede (ATM, Ethernet, Gigaethernet, ...);
- protocolos de comunicação (TCP, UDP, RTP, RSVP, ...) ; e
- meio de transmissão (fibra ótica, par trançado, microondas, ondas de rádio, raios infravermelhos, ...).

O algoritmo de compressão influencia diretamente a qualidade percebida pelo usuário final, uma vez que muitos deles foram projetados já tendo-se em mente aplicações e exigências de largura de banda e processamento específicas, traduzidas em limitações na qualidade de apresentação. Alguns algoritmos foram projetados para oferecer alta qualidade, exigindo, contudo, grande largura de banda e poder de processamento. Por causa dessas exigências, muitas vezes seu uso é viável apenas para a aplicações de reprodução de mídias armazenadas localmente, em disco ou CD-ROM. Outros algoritmos, como o H.261 e H.263, foram projetados especificamente para aplicações de videoconferência, exigindo poucos recursos mas oferecendo uma qualidade de apresentação baixa. De maneira similar, o formato G.723 foi projetado para aplicações de telefonia, cuja qualidade do áudio é baixa mas as exigências em termos de largura de banda também são baixas.

Em geral, quanto mais elevada é a qualidade desejada, maior será o tamanho do arquivo (para vídeo e/ou áudio armazenado) e maiores serão as exigências em termos de poder de processamento, memória RAM e largura de banda. Isso, contudo, não é uma regra: alguns algoritmos de compressão exigem pouco poder de processamento mas muita largura de banda (caso do M-JPEG) ou vice-versa. Assim, quando da seleção do algoritmo de compressão, é importante levar em conta as características e exigências do algoritmo. Por exemplo, se o fluxo multimídia deve ser transmitido via WWW, um aspecto que necessita atenção especial são as exigências da largura de banda. O uso de algoritmos de compressão com exigências altas em temos de recursos computacionais e/ou largura de banda em um SMD onde tais recursos são escassos refletir-se-á, necessariamente, numa baixa qualidade de apresentação.

Outro parâmetro de contexto que tem forte influência sobre a qualidade é a presença de codificadores implementados via "hardware" nos sistemas finais emissores, para aplicações de dados ao vivo. A compressão de quadros de vídeo é um processo que exige muito poder de processamento e se executada em tempo real via "software", limita a freqüência de quadros a valores inferiores àqueles proporcionados no caso da compressão via "hardware". O processo

Formato	Conteúdo	Qualidade	Exigências de Processamento	Exigências de Largura de Banda
Cinepak	AVI QuickTime	média	baixa	alta
MPEG-1	MPEG	alta	alta	alta
H.261	AVI RTP	baixa	média	média
H.263	QuickTime AVI RTP	média	média	baixa
JPEG	QuickTime AVI RTP	alta	alta	alta
Indeo	QuickTime AVI	média	média	média

Tabela 2.1: Formatos comuns de vídeo.

Formato	Conteúdo	Qualidade	Exigências de Processamento	Exigências de Largura de Banda
PCM	AVI QuickTime WAV	alta	baixa	alta
μ-Law	AVI QuickTime WAV RTP	baixa	baixa	alta
ADPCM (DVI, IMA4)	AVI QuickTime WAV RTP	média	média	média
MPEG-1	MPEG	alta	alta	alta
MPEG Layer3	MPEG	alta	alta	média
GSM	WAV RTP	baixa	baixa	baixa
G.723.1	WAV RTP	média	média	baixa

Tabela 2.2: Formatos comuns de áudio.

de descompressão de quadros de vídeo, por outro lado, mesmo se executado via "software", permite freqüências de quadros altas, similares àquelas obtidas nas transmissões regulares de TV ou em vídeo-tapes (em torno de 25 fps).

As Tabelas 2.1 e 2.2[9] identificam algumas das características dos algoritmos de compressão de vídeo e áudio mais comuns. Nestas tabelas, as colunas representam o algoritmo de compressão usado, o conteúdo do arquivo[10], o poder computacional necessário para uma boa apresentação no formato especificado e a velocidade de transmissão necessária, respectivamente.

Outros parâmetros de contexto que limitam a qualidade são a tecnologia de rede, protocolos de comunicação e meio de transmissão, especialmente em relação ao tempo de resposta. Contudo, neste trabalho o enfoque, em termos de parâmetros de contexto, resumir-se-á ao algoritmo de compressão e ao codificador, já que estes possuem características que podem ser exploradas por mecanismos de adaptação de QoS.

[9] http://java.sun.com/products/java-media/jmf/2.1/guide/

[10] A expressão "conteúdo" é usada ao invés de "tipo de arquivo" porque muitas vezes os dados das mídias são obtidos em tempo real, não sendo obtidos a partir de arquivos.

2.5.2 Parâmetros Controláveis

Os *parâmetros controláveis* são aqueles parâmetros de QoS que são acessíveis para a alteração de seus valores durante a sessão.

O conjunto dos parâmetros controláveis contém, principalmente, parâmetros de QoS da camada de aplicação, destacando-se:

- freqüência de quadros;
- freqüência de amostras de áudio;
- resolução espacial; e
- profundidade do pixel (número de bits para representação da cor).

Geralmente, o processo de adaptação de QoS culmina exatamente com a alteração nos valores desses parâmetros.

Além dos parâmetros acima, existe também uma série de parâmetros usados pelos algoritmos de compressão que são acessíveis e podem ser controlados via codificador ou decodificador. Esses parâmetros têm uma influência direta na fidelidade a imagem, ou seja, eles são parâmetros que, conforme o valor, tornam a imagem digital mais ou menos próxima da imagem real. Tais parâmetros influenciam também a taxa de compressão obtida e, conseqüentemente, as necessidades de largura de banda. No caso de algoritmos de compressão baseados em DCT, há um tipo de degradação representada pelas perdas introduzidas pelo processo de quantização. Em geral, quanto maior o coeficiente de quantização, maiores serão as perdas, visualizadas na forma de áreas borradas, manchas ao redor das bordas (chamadas *ruído de mosquito* - "mosquito noise") e imagem quadriculada (Verscheure et al. 1998).

Na camada do usuário, são parâmetros controláveis a qualidade do som e a qualidade da imagem, já que ambas são influenciadas diretamente pelos valores dos parâmetros da camada da aplicação mencionados anteriormente.

2.5.3 Parâmetros Não-Controláveis

Os *parâmetros não-controláveis* são parâmetros de QoS cujos valores podem mudar de maneira incontrolável durante a sessão. Os parâmetros não-controláveis são, na verdade, resultado das perturbações externas introduzidas no SMD. Os valores desses parâmetros não podem ser determinados *a priori*, e podem, ainda, variar de maneira brusca durante a transmissão e/ou

apresentação dos dados da aplicação. O conjunto de parâmetros não-controláveis tem como principais origens a carga da rede e a carga dos processadores.

Na camada do usuário, destaca-se como parâmetro não-controlável o tempo de resposta da aplicação.

Na camada da aplicação, são relevantes os seguintes parâmetros não-controláves:

- taxa de perdas de quadros;
- taxa de perdas de amostras de áudio;
- atraso fim-a-fim; e
- variação no tempo de resposta.

Na camada do sistema, os principais parâmetros não-controláveis são o tempo de processamento das tarefas e a taxa de perdas de "deadlines".

Na camada de comunicação, destacam-se os seguintes parâmetros não-controláveis:

- atraso;
- variação do atraso da rede;
- taxa de ocupação de "buffers";
- taxa de perdas de unidades de transporte; e
- taxa de erro de bits.

Os parâmetros não-controláveis podem ser usados pelos mecanismos de adaptação como *variáveis de realimentação* utilizadas para monitorar a carga do sistema ou a qualidade a apresentação.

2.5.4 Influência dos Parâmetros de Contexto, Controláveis e Não-Controláveis sobre a Qualidade da Apresentação

A qualidade da apresentação de uma aplicação multimídia é influenciada diretamente pelos valores dos parâmetros controláveis, como a freqüência de quadros ou amostras de áudio. Contudo, ela é limitada pelos parâmetros de contexto que determinam o intervalo de valores possíveis para os parâmetros controláveis. O poder de processamento do sistema final receptor, por exemplo,

determina o tempo de processamento das tarefas relacionadas à descompressão, o que limita o período das tarefas e, conseqüentemente, a freqüência de quadros.

Outros parâmetros que exercem influência sobre a qualidade são:

- o meio usado para transmissão, já que em redes sem fio ("wireless") a taxa de erros de bits não é desprezível como naquelas redes cujo meio físico é fibra ótica, o que influencia nas taxas de perdas de quadros e amostras de áudio;
- protocolo de comunicação, que define a quantidade de informação contida em uma unidade de transporte, influenciando na quantidade de amostras de áudio que podem ser perdidas com a perda de um pacote;
- prioridade das conexões, que define a ordem de retirada dos pacotes dos "buffers", influenciando no atraso fim-a-fim; e
- prioridade das tarefas de compressão/descompressão, que pode representar mais ou menos perdas de "deadlines".

Contudo, em termos de adaptação de QoS, o interesse maior recai sobre a influência das perturbações sobre os parâmetros não-controláveis, já que é esta a concernência da adaptação de QoS.

A mudança de valores dos parâmetros não-controláveis se propaga sobre os parâmetros controláveis, influenciando, assim, a qualidade da apresentação conforme pode ser visto nas relações de dependência mostradas no grafo da Figura 2.3. Tais dependências serão comentadas a seguir.

Uma carga da rede alta aumenta o valor da taxa de ocupação de "buffers" dos nós intermediários da rede. Quando essa taxa alcança 100%, pacotes começam a ser descartados. Além de aumentar o valor da taxa de perdas de pacotes, uma carga de rede alta faz com que os pacotes aguardem mais tempo nos "buffers", aumentando o atraso e a variação do atraso da rede, percebida pelo usuário final como uma variação do tempo de resposta da aplicação, parâmetro fundamental em aplicações interativas. Na camada da aplicação, as perdas de pacotes implicam em perdas de quadros e/ou amostras de áudio. Essas perdas são percebidas pelo usuário final na qualidade da imagem e do som sob a forma de estagnação e/ou deformação da imagem, lapsos no som e "dessincronização" entre o áudio e o vídeo.

O aumento do atraso, por sua vez, muitas vezes implica no descarte de quadros de vídeo ou amostras de áudio nos sistemas finais receptores devido às restrições temporais de apresentação

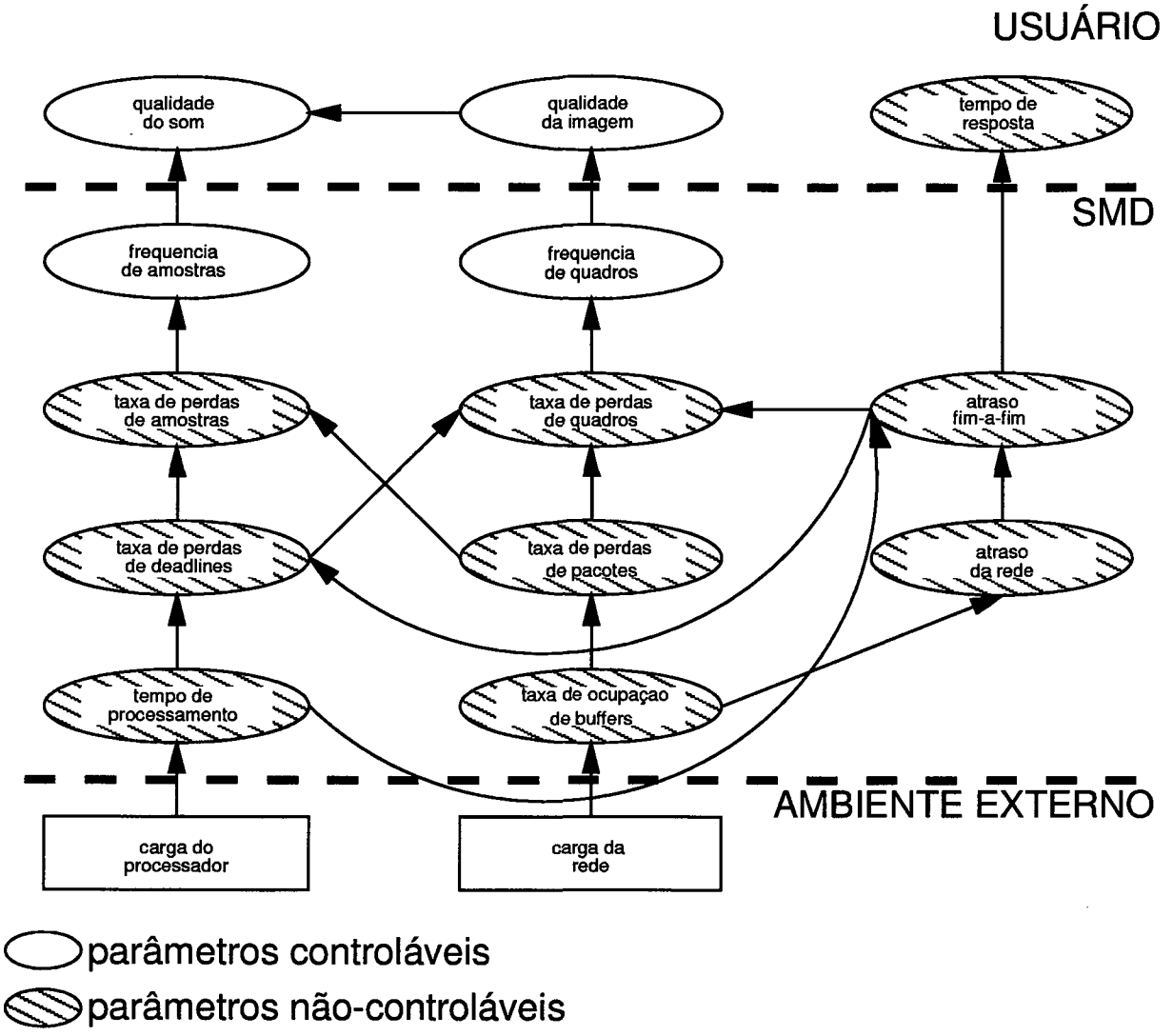

Figura 2.3: Grafo de dependências dos parâmetros de QoS.

desses dados. Tal problema ocorre também quando a carga do processador é alta, o que causa perda de "deadlines" da tarefa de descompressão com o conseqüente aumento das taxas de perdas de quadros de vídeo e amostras de áudio. Uma carga de processador alta também pode levar a tarefa de descompressão a aguardar mais tempo pelo uso do processador, alterando assim o valor do atraso e, conseqüentemente, o tempo de resposta.

Finalmente, a degradação da imagem pode implicar na dificuldade de percepção do som: se por um lado é evidente que a informação visual durante a fala (movimento dos lábios do interlocutor) pode auxiliar na compreensão do som, principalmente em condições anormais (qualidade de áudio pobre, ambiente barulhento etc.), por outro lado é sabido que a imagem pode também alterar a percepção do som através de um fenômeno conhecido como Efeito McGurk

Escala de Qualidade de Som	
Qualidade da fala	**Escore**
Excelente	5
Boa	4
média	3
Pobre	2
Ruim	1
Escala de Esforço de Escuta	
Esforço necessário para entender o significado das sentenças	**Escore**
Nenhum esforço necessário	5
Atenção necessária mas sem esforço significativo	4
Esforço moderado	3
Esforço considerável exigido	2
Sem entendimento do significado a despeito do esforço	1
Escala de Dificuldade de Conversação	
Você ou seu parceiro tiveram alguma dificuldade para falar ou ouvir durante a conexão	**Escore**
Sim	1
Não	0

Tabela 2.3: Escalas para atribuição de qualidade de som.

(McGurk e MacDonald 1976). Por exemplo, o som "ba" combinado com o movimento labial do som "ga" resulta na percepção de "da". Assim, uma qualidade de imagem pobre pode prejudicar a correta percepção do som.

A obtenção de uma função que permita quantificar a propagação de valores entre os parâmetros de QoS é tarefa complexa pois envolve outros fatores e parâmetros de QoS. O padrão do grupo de quadros (GOP) usado pelos codificadores de vídeo, por exemplo, pode fazer com que a perda de um pacote ocasione a perda de vários quadros. Se o GOP é formado apenas por quadros do tipo I, um pacote perdido ou alterado representará a perda de um único quadro; contudo, em um GOP com padrão $IPBBBBBB$, por exemplo, a perda de um pacote poderá representar a perda de oito quadros, se o pacote perdido contiver informações de um quadro do tipo I.

2.6 Avaliando a Qualidade

No Capítulo 1 foi dito que o objetivo deste trabalho é definir um mecanismo de adaptação de QoS que possibilite que a qualidade das aplicações multimídia mantenha-se dentro das expectativas do usuário mesmo diante de flutuações da carga da rede. Para que tal objetivo possa ser atingido, é necessário, então, que o mecanismo possa avaliar quando a qualidade está abaixo das expectativas do usuário. De maneira mais ampla, o mecanismo de adaptação de QoS deve estar apto a *comparar* diferentes qualidades, para que ele possa, diante de um determinado contexto, selecionar sempre a melhor. Essa necessidade de comparação de diferentes qualidades

traz consigo uma necessidade de uma avaliação prévia dessas diferentes qualidades, realizada através de algum método. A seguir, serão apresentados e discutidos alguns métodos usualmente utilizados para avaliação da qualidade.

Os métodos de avaliação da qualidade de áudio e vídeo mais comuns são aqueles recomendados pela ITU. Tais recomendações - ITU-T e ITU-R - endereçam testes ou entrevistas sobre grupos de usuários visando a atribuição de uma qualidade subjetiva para transmissões de áudio sobre redes de telefonia e vídeo sobre sistemas de televisão, respectivamente. Uma série de recomendações ITU-T também endereçam a atribuição de qualidade subjetiva para aplicações multimídia.

Para a atribuição de qualidade do som, a escala recomendada tanto para testes de só-escuta quanto conversação é uma escala de cinco pontos comumente conhecida como *escala de qualidade de som* (Int 1996). Essa escala atribui pontos de 1 a 5 para cada um dos termos subjetivos - excelente ("excellent"), boa ("good"), média ("fair"), pobre ("poor") e ruim ("bad") - usados nas entrevistas. O resultado médio obtido ("mean opinion score" ou MOS) é atribuído à amostra avaliada.

Os testes de só-escuta atribuem qualidade através de uma escala de esforço de escuta ("listening effort scale"); nos testes de conversação, uma escala de dificuldade binária segue a escala de qualidade. As escalas usadas para avaliação da qualidade do som são mostradas na Tabela 2.3.

Para a atribuição da qualidade da imagem (Tabela 2.4) são usados métodos de estímulo único ranqueados através de uma escala de qualidade ou escala de danos ("impairment"); uma escala contínua de qualidade de duplo estímulo ("double-stimulus continuous quality scale" - DSCQS, Figura 2.4) ou uma escala de danos de duplo estímulo é usada para efetuar comparações de um vídeos de teste com uma referência (Int 2000). A diferença entre os dois representa o dano. Geralmente, os vídeos são exibidos simultaneamente para o avaliador, que não é informado sobre qual deles representa a referência. A forma usada para votar contém cinco adjetivos (os mesmos usados para análise da qualidade do som). Contudo, os escores podem situar-se me valores de 0 a 100, onde 0 representa um qualidade ruim ("bad") e 100 excelente ("excellent"). A diferença dos escores do vídeo de teste e da referência situa-se também em um intervalo de 0 a 100, onde 0 e 100 representam, respectivamente, um baixo e um alto nível de danos. Podem ocorrer valores negativos, se os avaliadores consideram a qualidade do vídeo de teste melhor do que a referência. No método de avaliação de qualidade baseado em DSCQS, a escala de danos

Escala de Qualidade de Imagem	
Qualidade da Imagem	**Escore**
Excelente	5
Boa	4
média	3
Pobre	2
Ruim	1
Escala de Esforço de Visão	
Dano na Imagem	**Escore**
Imperceptível	5
Perceptível, mas não perturbador ("annoying")	4
Levemente perturbador	3
Perturbador	2
Muito perturbador	1
Escala Contínua de Qualidade de Duplo Estímulo (DSCQS)	
Vide Figura 2.4	

Tabela 2.4: Escalas para atribuição de qualidade de imagem.

da imagem é utilizada para avaliar uma Para atribuição de qualidade audiovisual, finalmente, a ITU recomenda uma metodologia baseada em testes de opinião. Nessa metodologia, uma escala de cinco pontos é usada para atribuir a qualidade do vídeo e áudio, individualmente, bem como a qualidade como um todo.

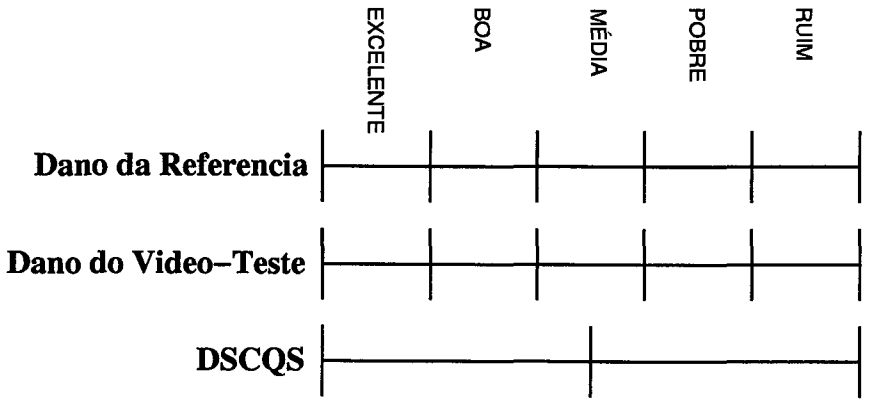

Figura 2.4: Escala contínua de qualidade de duplo estímulo.

Muitos autores ((Watson e Sasse 1998), (Jones e McManus 1996) etc.), contudo, argumentam que tais escalas e testes são inadequados para avaliação de comunicações multimídia em tempo real em decorrência dos seguintes aspectos:

1. rótulos e expressões utilizados: as recomendações ITU-T e ITU-R foram concebidas para a atribuição de qualidade subjetiva de sistemas de telefonia e televisão face a pequenas distorções, e não para comparar sistemas diferentes que oferecem qualidades diferentes, como NTSC, PAL, VHS, SQDTV e HDTV, para vídeo, ou telefonia, FM e CD

(monofônico e estereofônico), para áudio. Um áudio com qualidade de telefonia, por exemplo, jamais seria classificado como "excelente" se comparado a um áudio com qualidade de CD, mesmo na ausência de qualquer tipo de perturbação. As recomendações ITU-T e ITU-R também não foram concebidas visando comparar as qualidades resultantes após o uso de diferentes algoritmos de compressão ou a qualidade resultante do uso de diferentes combinações de valores de parâmetros de QoS;

2. universo de opções: o uso de um conjunto discreto e pequeno de opções para qualificação também contribui para que as respostas concentrem-se nos pontos mais baixos da escala. Mesmo no caso do DSCQS, que permite que os avaliadores escolham valores intermediários entre os rótulos, testes realizados (Aldrifge et al. 1995) mostraram que os avaliadores, em geral, se sentem inibidos em selecionar o limite superior da escala, preferindo posicionar sua opinião entre os pontos "boa" e "média";

3. tradução de rótulos e expressões: as escalas recomendadas pela ITU foram definidas utilizando adjetivos e expressões da língua inglesa. Além da subjetividade intrínseca a elas, a tradução para outras línguas não é trivial, não permitindo que os resultados obtidos em um país de língua inglesa possam ser generalizados para todo o mundo;

4. ordem de apresentação do material de testes: a ordem em que os clipes para avaliação são exibidos pode influenciar na opinião dos avaliadores. Jones e Atkinson do "Institute for Telecommunication Sciences" (ITS; EUA) mostram em (Jones e Atkinson 1998) os resultados de experimentos audiovisuais subjetivos. Foram usadas seis cenas processadas em 8 diferentes configurações, obtidas a partir da combinação de um algoritmo de compressão (H.261) com duas diferentes resoluções (CIF - 352×288 pixels - e QCIF - 176×144 pixels) e três diferentes algoritmos de compressão de áudio (G.711, G.722 e G.728), além de um vídeo analógico padrão NTSC e um algoritmo de compressão de áudio-vídeo proprietário, resultando em um total de 48 clipes. As avaliações foram realizadas sobre MOS's obtidos a partir da opinião de 18 avaliadores leigos em relação a três sessões: somente áudio, somente vídeo e audiovisual. Dentre outros resultados, os escores obtidos nas sessões contendo somente vídeo ou áudio-vídeo, quando apresentadas por último, foram mais altos do que quando o mesmo material era apresentado na primeira ou segunda sessão; e

5. desconsideração de variáveis: a metodologia de avaliação baseada em MOS desconsidera

uma série de variáveis que podem influenciar nos resultados. A referência utilizada (por exemplo, o clipe com qualidade máxima) pode levar a diferentes resultados; o estado emocional bem como o grau de conhecimento dos avaliadores podem produzir julgamentos completamente distintos: um usuário consciente do significado dos parâmetros de QoS que compõem o clipe avaliará a qualidade de forma diferente de um usuário totalmente leigo. A natureza do clipe também exerce influência sobre o julgamento. Aldridge et al. observaram em seus experimentos que alguns avaliadores distraíram-se de sua tarefa de atribuição da qualidade em virtude do conteúdo do clipe exibido (Aldrifge et al. 1995).

Além das restrições descritas acima, muitos autores questionam a validade dos testes pelo fato deles não reproduzirem com fidelidade o ambiente encontrado em um SMD. As perturbações existentes em um SMD, conforme visto na Seção 2.5.3, ocasionarão reduções na freqüência de quadros e lapsos de som fazendo com que a qualidade realmente obtida dificilmente possa ser descrita como "excelente". Também expressões como "perceptível", "perturbador" e "muito perturbador" podem ser inadequadas para análise de qualidade em SMD's por terem sido usadas, originalmente, para qualificar o resultado de *pequenas distorções no vídeo*. Especificamente no caso da ITU-T, a escala de dificuldade de conversação, se usada para avaliar o áudio em um SMD, provavelmente resultará numa concentração expressiva de respostas "sim", já que mesmo uma taxa de perdas de pacotes baixa causa uma dificuldade na escuta. Além disso, é possível que um áudio com qualidade de telefonia com baixa taxa de perdas seja taxado como "excelente" enquanto um áudio com qualidade de CD seja taxado como "ruim", se reproduzido com uma alta taxa de perdas.

Em (Watson e Sasse 1996), são mostrados os resultados de testes de avaliação da qualidade de áudio realizados em um ambiente que, através da geração aleatória de perdas de pacotes, simulava a Internet. Os resultados mostraram que as respostas de testes tendem, de fato, a considerar não só a qualidade propriamente dita, mas também degradações introduzidas pelo ambiente, concentrando-se nos pontos (termos) mais baixos ("média", "pobre" e "ruim"). Os resultados mostraram também que, além das perdas, o método usado para compensá-las influenciava os resultados da avaliação.

No que concerne à adaptação de QoS, talvez a principal limitação não esteja nos métodos de avaliação de qualidade propostos e sim nos testes realizados, já que nenhum deles é voltado para a comparação de vídeos comprimidos com diferentes qualidades obtidas a partir de diferentes combinações dos valores dos parâmetros do codificador. Testes tendo como objetivo esse tipo

de comparação forneceriam subsídios que habilitariam os mecanismos de adaptação de QoS a escolher a melhor combinação de valores de parâmetros de QoS (em termos de qualidade) diante do contexto corrente do SMD.

2.7 Resumo e Discussão

Neste capítulo, foram apresentados vários conceitos para QoS encontrados na literatura, sendo que a definição de QoS fornecida por Vogel et al. em (Vogel et al. 1995), que abrange atributos das várias entidades do SMD (e não somente aqueles do sistema de comunicação), será a adotada neste trabalho.

Também foi descrita a visão arquitetural do SMD a ser utilizada neste trabalho e na qual o SMD é dividido em cinco camadas (usuário, aplicação, sistema, comunicação e dispositivos).

Uma vez que o controle de QoS não pode ser desvinculado da natureza da aplicação, pois ela determina a importância relativa e limites de valores dos parâmetros de QoS, neste capítulo foi proposta uma taxinomia para aplicações multimídia voltada para aspectos de QoS. Os critérios usados para dividir as aplicações foram manutenção da QoS, interatividade, fonte de obtenção dos dados, conteúdo dos quadros, sincronização entre os fluxos e distribuição dos dados. Tais critérios não são mutuamente exclusivos: uma mesma aplicação pode ser enquadrada em várias categorias da taxinomia proposta.

Neste capítulo foi proposta, ainda, a divisão dos parâmetros de QoS que influenciam a qualidade percebida pelo usuário final em três grupos: parâmetros de contexto, parâmetros controláveis e parâmetros não-controláveis. O primeiro grupo é representado por parâmetros relacionados ao ambiente onde a aplicação multimídia distribuída está sendo executada, não sendo alvo da adaptação de QoS mas sim de uma configuração preliminar. Estes determinam, em conjunto com a natureza da aplicação, os limites de atuação de um mecanismo de adaptação. Os parâmetros controláveis são aqueles que serão modificados pelo mecanismo de adaptação de QoS. O terceiro grupo é representado pelos parâmetros de QoS cujos valores são decorrência das perturbações do sistema. Em síntese, o mecanismo de adaptação de QoS atua sobre os parâmetros controláveis, dentro dos limites de valores impostos pelos parâmetros de contexto e pela natureza da aplicação, em resposta às mudanças causadas pelos parâmetros não-controláveis que agem sobre o SMD e que influem indiretamente sobre os valores dos parâmetros controláveis.

Por fim, foram apresentadas algumas das formas de avaliação de qualidade mais utilizadas. A avaliação da qualidade permite que diferentes qualidades sejam comparadas, permitindo que o mecanismo de adaptação, diante de um dado contexto em termos de disponibilidade de recursos, selecione a melhor qualidade possível. Atualmente, as metodologias mais utilizadas para avaliação de qualidade de áudio e vídeo são aquelas propostas pela ITU, a despeito das críticas de alguns autores às mesmas. Contudo, na maior parte das avaliações realizadas e disponíveis na literatura, as análises têm tido como escopo a comparação de diferentes paradigmas de vídeo (SQDTV, HDTV, sistema PAL, sistema NTSC, vídeo-tape, ...) e áudio (telefonia, CD, FM, ...) ou a avaliação da qualidade audiovisual frente às perturbações do SMD, existindo uma lacuna no que se refere à avaliação e comparação de diferentes combinações de valores de parâmetros de QoS. Com o intuito de preencher essa lacuna, foi desenvolvido no âmbito do Laboratório de Controle e Microinformática da Universidade Federal de Santa Catarina uma ferramenta para realização de testes subjetivos de qualidade de vídeo. Os testes podem ser realizados com clipes codificados com diferentes algoritmos de compressão e com diferentes qualidades, obtidas através da mudança de parâmetros do codificador (coeficiente de quantização e freqüência de quadros de vídeo, inicialmente). Os resultados das avaliações, quantificados através de MOS's, fornecerão dados estatísticos sobre a aceitação e tolerância da perda de qualidade. Tais dados permitirão a construção de funções graus de qualidade para diferentes algoritmos de compressão.

No próximo capítulo, definir-se-á uma função para a representação de qualidade. Nessa função, diferentes qualidades são representadas por diferentes combinações de valores de parâmetros de QoS da camada da aplicação. Uma métrica, cujo valor é obtido através de avaliações, é usada para comparar essas diferentes qualidades.

Capítulo 3

FUNÇÃO GRAU DE QUALIDADE

3.1 Introdução

No Capítulo 2 foi vista uma série de aspectos relacionados à qualidade (parâmetros de QoS envolvidos e sua influência sobre a qualidade, formas de avaliação, ...). Contudo, um aspecto que não têm merecido a devida atenção na maior parte dos trabalhos relacionados é a definição de uma *forma de representação da qualidade* que permita a comparação de diferentes qualidades e que leve em conta, prioritariamente, o ponto de vista do usuário final. Isso implica em uma forma de representação da qualidade que a considere como um todo, envolvendo vários parâmetros de QoS. No que diz respeito à adaptação de QoS, a ausência da representação da qualidade como um fenômeno multidimensional tem sido percebida na forma de propostas que procuram sempre otimizar um único parâmetro de QoS, o que torna o processo de adaptação distante do usuário.

Em decorrência das limitações encontradas na literatura no que concerne à forma de representação da qualidade, neste trabalho é proposta uma representação baseada em tuplas onde cada elemento representa uma dimensão da qualidade, ou seja, cada tupla representa uma combinação de valores de parâmetros de QoS. Além disto, visando ordenar ou comparar essas tuplas (também sob o ponto de vista do usuário final), é proposto neste trabalho o conceito de *grau da qualidade*, uma métrica obtida através de uma função referenciada como *função grau de qualidade*. Essa métrica e a função usada para obtê-la serão descritas a seguir após a apresentação de alguns trabalhos correlatos.

3.2 Trabalhos Relacionados

Na literatura, existem alguns trabalhos que contemplam a definição de uma forma de comparação de valores de parâmetros de QoS individuais. Outros trabalhos associam a combinações de parâmetros de QoS algum tipo de medida. Contudo, os parâmetros-alvo não são necessariamente da camada da aplicação (ou seja, aqueles parâmetros percebidos pelo usuário) e a medida associada não é necessariamente relacionada à qualidade de apresentação.

Em (Krasic e Walpole 1999), é descrita uma abordagem para o mapeamento de requisitos de QoS para políticas de escalamento ("scaling´´) de consumo de recursos. Tal abordagem é baseada no uso de várias *funções utilidade*. Uma função utilidade é uma função unidimensional que mapea valores de um parâmetro de QoS da camada de aplicação para uma utilidade subjetiva pertencente ao intervalo [0,1]. A função deve ser obtida a partir das preferências dos usuários, mas os autores fazem algumas premissas a respeito de seu comportamento. Ela deve possuir dois pontos limites, correspondendo às utilidades 0 e 1. Tais pontos não correspondem, necessariamente, aos valores das qualidades máxima e mínima do parâmetro considerado, mas sim a um valor acima do qual o usuário não percebe mais melhora na qualidade e um ponto abaixo do qual os valores são igualmente ruins em termos de qualidade para o usuário. Entre esses dois pontos, a função é contínua e monotônica crescente. Em (Richards et al. 1998), é proposta uma função semelhante, porém modelada matematicamente a partir de uma série de premissas relacionadas à percepção do usuário à variação dos valores de parâmetros de QoS individuais da camada de aplicação.

Em (Bocheck et al. 1999), é descrita uma função chamada *curva de utilidade de largura de banda* ("bandwidth utility curve"). A curva de utilidade é gerada automaticamente através do uso uma função quadrática, linear ou discreta, associando pontos representando largura de banda a pontos de um intervalo de 1 a 5 (MOS's) que representam diferentes graus de qualidade subjetiva de vídeo. A construção automática da curva de utilidade de largura de banda parte da premissa de que uma maior largura de banda *sempre* representará maior qualidade. Tal premissa, contudo, não é totalmente verdadeira: um fluxo de vídeo comprimido com o algoritmo H.263, por exemplo, se emitido a 30 fps e com fator de quantização igual a 30 exige mais largura de banda do que se emitido a 15 fps e com fator de quantização 14. Contudo, no primeiro caso, a qualidade da imagem é muito mais baixa do que no segundo caso, já que um fator de quantização igual a 30 torna os elementos que compõem a imagem praticamente indistinguíveis entre si. Por outro lado, um fator de quantização igual a 14 resulta em uma imagem com qualidade bastante

razoável.

Em (Abdelzaher e Shin 1998) é descrito um mecanismo de adaptação de QoS baseado em um contrato entre as partes (sistemas finais emissor e receptor). Tal contrato especifica os requisitos de QoS através das combinações de valores de três parâmetros da camada do sistema: período das tarefas (considerado como sendo o inverso da freqüência de emissão), tamanho de "buffer" e o tamanho da unidade de dado a ser processada a cada período. A cada combinação é associada uma *recompensa* ("reward"), uma medida arbitrária de desempenho (por exemplo, para uma aplicação de controle de vôo, citada no trabalho, a recompensa é a probabilidade de sucesso da missão) ou, pelo lado do provedor do serviço, o custo do serviço que o cliente pagará para aqueles valores de parâmetros de QoS. Assim, o significado das recompensas varia de acordo com a aplicação. Porém, a forma como elas são estimadas não é endereçada no trabalho.

Em (Hull et al. 1995), é descrito um modelo de QoS onde a aplicação multimídia é representada através de várias tarefas produtor-consumidor e a qualidade de saída de uma tarefa é dada por uma *função recompensa* ("reward function") cujos argumentos são a qualidade de entrada da tarefa (ou seja, os valores dos parâmetros de QoS usados por ela) e a quantidade de recursos alocada. Todas as tarefas têm uma função recompensa conhecida que é fornecida pelo "desenvolvedor" da aplicação, produzida por alguma ferramenta do sistema ou dada por uma função "default". Uma relação de ordem parcial, chamada *função valor*, especifica como uma tarefa consumidora compara as diferentes qualidades representadas pelas diferentes combinações de valores dos parâmetros de QoS de saída de sua tarefa produtora. A função valor para a tarefa de apresentação dos dados multimídia poderia ter alguma semelhança com a função grau de qualidade proposta neste trabalho. Entretanto, há uma série de aspectos vagos em relação a ela: os autores não definem uma metodologia para sua obtenção, quais parâmetros de QoS representam suas dimensões, qual a participação do usuário na sua construção etc.

Em (Fry et al. 1996) é descrito um mecanismo de adaptação de QoS direcionado para aplicações da "World Wide Web" que utiliza uma tabela para definir limites e prioridades de degradação. Essa tabela, referenciada como *caminho de degradação* (Vogel et al. 1995), é criada a partir das preferências dos usuários em relação aos parâmetros de QoS contidos em suas entradas (resolução espacial, algoritmo de compressão e freqüência de quadros). A cada combinação de parâmetros de QoS, é associada a taxa de transmissão máxima. O caminho de degradação não associa nenhuma medida arbitrária de qualidade às combinações: a ordem com

que as entradas estão distribuídas na tabela indica apenas uma precedência em termos de qualidade para as combinações de parâmetros de QoS para *um usuário em particular* (antes do início da transmissão, o usuário fornece suas preferências em relação aos parâmetros considerados através da atribuição de pesos).

De maneira geral, as propostas acima têm uma ou mais das seguintes limitações:

1. elas associam uma métrica (supostamente obtida a partir de entrevistas com usuários) a parâmetros de QoS individuais, desconsiderando a influência que um parâmetro de QoS exerce sobre a percepção de outros;
2. elas não mostram como é feita a associação de alguma métrica a combinações de parâmetros de QoS;
3. os parâmetros de QoS considerados não pertencem à camada da aplicação; e
4. a preocupação da representação é com o consumo de recursos e não com a qualidade.

3.3 A Função Grau de Qualidade

Uma aplicação multimídia distribuída pode ter suas exigências, em termos de recursos, adaptadas ao contexto corrente do SMD através da alteração dos valores de seus parâmetros de QoS controláveis. Entretanto, essa abordagem, *per si*, revela-se limitada e incompleta haja visto que várias combinações de valores de parâmetros de QoS com exigências de recursos similares podem representar qualidades totalmente distintas sob o ponto de vista do usuário. A definição de uma *função grau de qualidade* (Reis 2000) (Koliver et al. 2000 a) que associa um grau a cada combinação de valores de parâmetros de QoS disponibiliza um critério de seleção da melhor combinação diante do contexto corrente do SMD. A seguir, serão apresentados alguns conceitos e definições que permitirão construir tal função.

3.3.1 Definições

Seja ρ_i um parâmetro de QoS genérico da camada de aplicação e

$$\Omega_{\rho_i} = [\rho_{i_{min}}, \rho_{i_{max}}] \; (i = 1, 2, ..., n) \tag{3.1}$$

o domínio de valores de ρ_i. Seja também

$$L =< \rho_1, \rho_2, ..., \rho_n > \tag{3.2}$$

uma combinação de valores dos n parâmetros de QoS da camada de aplicação ou um *nível de QoS*. Assim, o conjunto de todos os níveis de QoS possíveis é obtido através do produto cartesiano dos n domínios Ω_{ρ_i}.

Visando diferenciar os níveis de QoS de acordo com a percepção do usuário, é definida uma métrica de qualidade, chamada *grau de qualidade* ($Q\hat{o}S$), arbitrariamente incluída no domínio $[0, 1]$. O valor de $Q\hat{o}S$ de um dado nível de QoS é obtido usando-se a *função grau de qualidade* ($\mathcal{QoS}$) (Koliver et al. 2000 b):

$$\mathcal{QoS} : \Omega_{\rho_1} \times \Omega_{\rho_2} \times ... \times \Omega_{\rho_n} \mapsto [0, 1]. \tag{3.3}$$

Para o nível de QoS

$$L_j =< \rho_{1_j}, \rho_{2_j}, ..., \rho_{n_j} >, \tag{3.4}$$

o grau de qualidade assume um valor

$$Q\hat{o}S_j = \mathcal{QoS}(< \rho_{1_j}, \rho_{2_j}, ..., \rho_{n_j}). \tag{3.5}$$

3.3.2 Comportamento

O real comportamento da função grau de qualidade só pode ser obtido a partir de uma construção alicerçada em dados obtidos a partir de entrevistas com usuários. Contudo, tal comportamento possui uma série de características esperadas baseadas em noções intuitivas a respeito da percepção do usuário em relação à qualidade. Dentre essas características, destacam-se:

1. para determinados níveis de QoS, a função $\mathcal{QoS}$ permanece invariável. Isso decorre do fato de dois ou mais níveis de QoS poderem ter o mesmo grau de qualidade, existindo, dentre outros, dois subconjuntos de níveis de QoS cujos elementos têm grau de qualidade 0 e grau de qualidade 1. Essa suposição é baseada em uma noção intuitiva em relação à percepção do usuário quanto à qualidade: existem subconjuntos de níveis de QoS cujos elementos, em termos de qualidade, são indistinguíveis entre si para o usuário final; um desses subconjuntos contém níveis de QoS cuja qualidade está aquém das expectativas

mínimas do usuário e outro contém níveis de QoS que representam a percepção máxima de qualidade por parte do usuário final;

2. em determinados intervalos, a mudança de QoS é bem mais brusca do que em outros. Essa suposição é baseada na noção intuitiva de que o usuário percebe mais a variação da qualidade quando um parâmetro de QoS tem seu valor levemente alterado para mais ou menos se o valor corrente desse parâmetro é baixo (em termos de qualidade). Por exemplo, a mudança da freqüência de quadros de 5 para 8 fps é mais sentida pelo usuário do que a mudança de 20 para 23 fps. Isso implica que quanto mais baixo for o valor de um parâmetro de QoS, maior será a alteração do grau de qualidade quando o valor desse parâmetro é alterado, mantendo-se os outros constantes; e

3. QoS tende a fornecer um valor baixo para o grau de qualidade se um parâmetro qualquer tende a seu valor mínimo (também em termos de qualidade), independentemente dos valores dos outros parâmetros. Essa suposição é baseada na noção intuitiva de que a qualidade é baixa se um parâmetro de QoS qualquer tem um valor muito baixo, mesmo que os valores dos outros parâmetros sejam altos, já que um parâmetro de QoS com um valor "ruim" é suficiente para que o usuário considere a qualidade baixa.

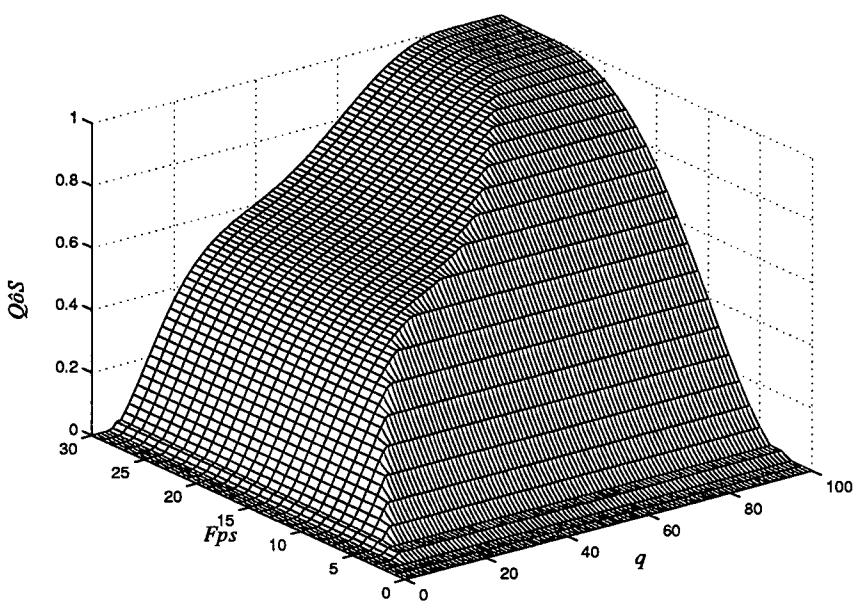

Figura 3.1: Grau de qualidade $Q\hat{o}S$ em função da freqüência de quadros e do fator de quantização.

Na Figura 3.1 (Koliver et al. 2000 a) é mostrado o comportamento da função $\mathcal{QoS}(< Fps, q >)$, onde Fps é a freqüência de quadros e q o fator de quantização. O domínio da de Fps é $\Omega_{Fps} = \{0, 1, 2, ...30\}$; o domínio de q (normalizado) é $\Omega_q = \{0, 1, 2, ...100\}$, onde 100 representa a mais alta qualidade e 0 a mais baixa. Tais valores referm-se à aplicação de videoconferência VIC (McCanne et al. 1996) moldado de acordo com as premissas assumidas anteriormente. A superfície foi construída utilizando-se uma metodologia baseada em técnicas de interpolação com pontos de inflexão arbitrados a partir de analogias (qualidade de videoconferência, qualidade de TV-PAL, qualidade de TV-NTSC) e dos dados levantados em (Wallace 1991) e (Reininger et al. 1998) que associam MOS's a valores de fator de quantização do algoritmo de compressão M-JPEG (aquele usado pela aplicação).

3.3.3 Obtenção da Função Grau de Qualidade

A despeito da função grau de qualidade poder ser moldada considerando-se as características apresentadas anteriormente, somente uma construção baseada em dados reais (isto é, dados obtidos a partir de entrevistas com usuários) resultará em uma função que, de fato, reflita o pensamento médio dos usuários. Nesta seção, são apresentadas duas metodologias para construção de $\mathcal{QoS}$ baseadas na opinião de usuários.

Abordagem 1: Avaliação Sobre Parâmetros de QoS

O primeiro passo para construção de $\mathcal{QoS}$ consiste na definição do conjunto de parâmetros de QoS a serem considerados. Em princípio, todos os parâmetros da camada da aplicação devem ser considerados, de modo que os níveis de QoS englobem os parâmetros relacionados à qualidade do som e imagem bem como o tempo de resposta.

O passo seguinte consiste na determinação do grau de qualidade associado a cada nível de QoS. Para tal, cada um dos n parâmetros é avaliado individualmente por um grupo de avaliadores que deve assistir clipes nos quais o valor de um único parâmetro ρ_i varia enquanto os outros $n - 1$ permanecem fixos, configurados com os valores que representam a melhor qualidade. Assim, há um clipe para cada valor de ρ_i. A avaliação dos clipes pode ser feita, por exemplo, através de MOS's. O processo repete-se para cada um dos n parâmetros considerados. A média dos resultados obtidos para cada amostra deve ser normalizada para valores entre 0 e 1 (o domínio de valores de $\mathcal{QoS}$). O resultado dessas avaliações permite a construção de n

funções utilidade. Seja v_{ρ_i} a função utilidade obtida para o parâmetro ρ_i, isto é,

$$v_{\rho_i} : \Omega_{\rho_i} \mapsto [0, 1]. \tag{3.6}$$

Para o nível de QoS

$$L_j =< \rho_{1_j}, \rho_{2_j}, ..., \rho_{n_j} >,$$

o grau de qualidade é obtido a partir da equação abaixo:

$$\mathcal{QoS}(< \rho_{1_j}, \rho_{2_j}, ..., \rho_{n_j} >) = min(v_{\rho_1}(\rho_{1_j}) \times \omega_{\rho_1}, v_{\rho_2}(\rho_{2_j}) \times \omega_{\rho_2}, ..., v_{\rho_n}(\rho_{n_j}) \times \omega_{\rho_n}), \tag{3.7}$$

onde ω_{ρ_i} ($\sum_{i=1}^{n} \omega_{\rho_i} = 1$) é o peso do parâmetro ρ_i na avaliação da qualidade como um todo. O valor dos pesos deve ser baseado em estudos que considerem aspectos fisiológicos e psicológicos relacionados à percepção das dimensões da qualidade, como aqueles encontrados em (Winkler 1999).

A estratégia para construção da função $\mathcal{QoS}$ descrita acimanunca permite que $\hat{QoS_j} = 1$. O grau de qualidade máximo possível de ser alcançado é

$$\hat{QoS_j} = \frac{1}{n} \tag{3.8}$$

se $v_{\rho_1}(\rho_{1_j}) = v_{\rho_2}(\rho_{2_{L_j}}) = ...v_{\rho_n}(\rho_{n_j}) = 1$ and $\omega_{\rho_1} = \omega_{\rho_2} = ... = \omega_{\rho_n}) = \frac{1}{n}$. Visando manter a definição da função $\mathcal{QoS}$ dada pela Equação 3.3, os resultados obtidos usando a estratégia definida pela Equação 3.7 devem ser normalizados no intervalo [0,1].

Abordagem 2: Avaliação Sobre Níveis de QoS

Na segunda abordagem para a determinação do grau de qualidade associado a cada nível de QoS, as avaliações são feitas diretamente sobre os níveis de QoS, ou seja, há um clipe correspondente a cada nível de QoS. A vantagem dessa abordagem é que os graus de qualidade realmente refletirão a opinião média dos usuários enquanto abordagem 1 eles serão modelados a partir de alguns dados reais. A desvantagem é que essa abordagem pode exigir que cada avaliador analise $\prod_{i=1}^{n} \mathcal{C}(\rho_i)$ clipes enquanto na abordagem 1 são necessários no máximo $\sum_{i=1}^{n} \mathcal{C}(\rho_i)$ clipes onde $\mathcal{C}(\rho_i)$ é o número total de níveis de QoS (a cardinalidade do conjunto $\Omega_{\rho_1} \times \Omega_{\rho_2} \times ... \times \Omega_{\rho_n}$)

Redução do Número de Clipes a Serem Avaliados

A avaliação da qualidade pode tornar-se um processo extremamente maçante e cansativo para o avaliador, conforme o número de amostras a serem avaliadas. A construção da função grau de qualidade da Figura 3.1, por exemplo, exigiria a avaliação de $31+101 = 132$ clipes utilizando-se a primeira abordagem ou $31 \times 101 = 3131$ clipes para a segunda abordagem. Considerando-se um mínimo de 10 segundos para cada clipe (conforme a recomendação da ITU), a primeira abordagem exigiria a atenção de um avaliador durante mais de 22 minutos enquanto a segunda exigiria mais de 8 horas. Tais valores poderiam aumentar drasticamente com o aumento de parâmetros de QoS considerados.

Uma possibilidade para reduzir o número de clipes a serem avaliados consiste no arbitramento de um subconjunto de amostras com a extensão dos resultados efetuada através de alguma técnica de interpolação.

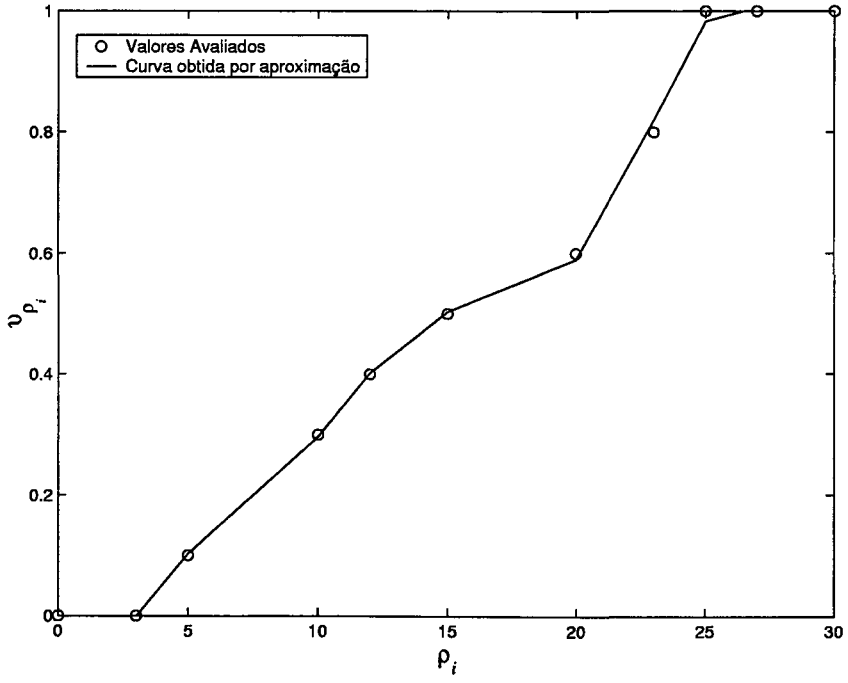

Figura 3.2: Função utilidade obtida por regressão linear.

No caso da abordagem 1, por exemplo, pode-se reduzir o número de clipes obtendo-se a função utilidade para os valores de parâmetros de QoS não analisados usando-se regressão linear simples (método dos mínimos quadrados). Seja, por exemplo, um parâmetro de QoS ρ_i de domínio

$$\Omega_{\rho_i} = \{0, 1, 2, ..., 29, 30\}.$$

Sejam também, os seguintes valores médios para a utilidade obtidos para o subconjunto (gerado aleatoriamente ou arbitrado) $\{0, 3, 5, 10, 12, 15, 20, 23, 25, 27, 30\} \in \Omega_{\rho_i}$ através de entrevistas com um grupo de avaliadores:

$$\upsilon_{\rho_i}(0) = 0.0,$$
$$\upsilon_{\rho_i}(3) = 0.0,$$
$$\upsilon_{\rho_i}(5) = 0.1,$$
$$\upsilon_{\rho_i}(10) = 0.3,$$
$$\upsilon_{\rho_i}(12) = 0.4,$$
$$\upsilon_{\rho_i}(15) = 0.5,$$
$$\upsilon_{\rho_i}(20) = 0.6,$$
$$\upsilon_{\rho_i}(23) = 0.8,$$
$$\upsilon_{\rho_i}(25) = 1.0,$$
$$\upsilon_{\rho_i}(27) = 1.0,$$
$$\upsilon_{\rho_i}(30) = 1.0.$$

Através de interpolação polinomial, é obtida a seguinte função utilidade:

$$\upsilon_{\rho_i}(\rho_i) = -0.0003 \times \rho_i^5 + 0.0043 \times \rho_i^4 - 0.0329 \times \rho_i^3 + 0.1317 \times \rho_i^2 - 0.1931 \times \rho_i + 0.0002. \quad (3.9)$$

Na Figura 3.2 é mostrada a curva obtida por regressão linear para o exemplo acima.

Na abordagem 2, pode-se obter também uma função matemática para $\mathcal{QoS}$ através de um subconjunto de níveis de QoS avaliados. Uma regressão linear múltipla é usada para a obtenção da função, dada pela equação

$$\mathcal{QoS} = c_1 \times \rho_1 + c_2 \times \rho_2 + ... + c_{n-1} \times \rho_{n-1} + c_n \times \rho_n + c_{n+1} \quad (3.10)$$

onde c_j $(j = 1, 2, ..., n, n+1)$ são os coeficientes obtidos por aproximação.

Representação Alternativa da Função Grau de Qualidade

Além da representação através de uma função matemática, conforme visto na seção anterior, $\mathcal{QoS}$ pode ser representada através de uma tabela onde cada registro é uma tupla do tipo

$$< \rho_1, \rho_2, ..., \rho_n, \hat{QoS} > .$$

Tal representação é mais adequada para o uso da função grau de qualidade para mecanismos de adaptação de QoS proposto neste trabalho, como será visto no Capítulo 5.

3.4 Resumo e Discussão

Neste capítulo foi apresentada uma função, referenciada como função grau de qualidade ou $\mathcal{QoS}$. O objetivo dessa função é associar a combinações de valores de parâmetros de QoS da camada da aplicação uma métrica que permita comparar essas combinações entre si em termos de qualidade, de acordo com a perspectiva do usuário. O uso de uma métrica ao invés de termos subjetivos permite uma gama muito maior de pontos de avaliação.

Ao contrário de outras propostas similares encontradas na literatura, para que a função realmente reflita a perspectiva do usuário, sua construção deve necessariamente ser embasada em dados obtidos a partir de alguma forma de avaliação de qualidade realizada por um grupo de pessoas. Uma modelagem matemática da percepção do usuário em relação à qualidade, a partir de certas premissas, é uma generalização que não contempla as especificidades de cada parâmetro de QoS.

Também, diferentemente de outras propostas, a função é multidimensional, *i.e.*, considera todos os parâmetros da camada de aplicação. Tal representação da qualidade parece mais próxima da realidade do que representações unidimensionais, já que o valor de um ou mais parâmetros de QoS pode ter influência na percepção do usuário em relação a outro parâmetro, conforme visto no Capítulo 2.

Funções que relacionam parâmetros de QoS individuais a alguma métrica de qualidade podem ser usadas para simplificar o processo de construção de $\mathcal{QoS}$. Contudo, para que a função resultante realmente considere a qualidade como um todo e realize uma ordenação dos níveis de QoS que reflita o pensamento médio do usuário final de maneira razoavelmente próxima, os parâmetros de QoS devem ser ponderados através da atribuição de pesos - baseados em estudos fisiológicos e psicológicos da percepção humana a parâmetros de QoS combinados - que representem a importância relativa do parâmetro na qualidade.

A função $\mathcal{QoS}$ pode ser usada por mecanismos de adaptação de QoS para avaliar a qualidade que os usuários da aplicação multimídia estão recebendo. Além disso, ela pode ser usada como critério para análise de desempenho dos mecanismos, como em (Koliver et al. 2000 a) e (Koliver e Farines 2001). Neste caso, a análise, ao invés de considerar parâmetros não-controláveis (os

parâmetros usualmente utilizados, como taxa de perdas de pacotes ou taxa de perdas de quadros) e distantes da ponto de vista do usuário final, considera como critério de desempenho o grau de qualidade. A inclusão de mais dimensões representando o *custo* associado a cada nível de QoS (em termos de consumo de recursos) disponibiliza um critério que permite aos mecanismos de adaptação selecionar a melhor combinação de valores de parâmetros de QoS ante o contexto do SMD. Essa possibilidade será discutida mais detalhadamente no Capítulo 5.

O próximo capítulo é dedicado à análise de mecanismos de adaptação de QoS de uma forma geral.

Capítulo 4

ADAPTAÇÃO DE QoS

4.1 Introdução

No Capítulo 3 foi visto que existem diversos parâmetros de QoS que modificam de forma incontrolável a qualidade da apresentação das aplicações multimídia distribuídas. Esses parâmetros, referenciados neste trabalho como parâmetros não-controláveis, são resultado das perturbações externas introduzidas no SMD, tendo como principais origens a carga da rede e a carga dos processadores. Os parâmetros não-controláveis podem levar outros parâmetros de QoS a valores fora de limites (em termos de qualidade de apresentação), diminuindo em muito a satisfação do usuário final. Essa situação conduz à necessidade da existência de mecanismos que adaptem a aplicação ao contexto do SMD, de forma que os efeitos negativos provocados pelas perturbações e suas conseqüências sejam controlados e minimizados. Este capítulo é dedicado à análise geral desses mecanismos, aqui referenciados como *mecanismos de adaptação de QoS*. A análise dos mecanismos de adaptação de QoS será realizada a partir das respostas às seguintes questões:

1. Por que realizar a adaptação?
2. O que é necessário para a adaptação?
3. Quando será realizada a adaptação?
4. Onde ocorrerá a adaptação?
5. Que parâmetros serão adaptados?
6. Como será feita a adaptação?

4.2 Conceitos de Adaptação de QoS

4.2.1 Definição

Neste trabalho, será assumido que *a adaptação de Qos é a atividade de atuar sobre o fluxo gerado pela aplicação multimídia distribuída, tendo como principal objetivo ajustar, de forma suave e controlada, a qualidade oferecida aos usuários finais frente à variação da disponibilidade de largura de banda da rede.* Essa definição restringe a adaptação de QoS à reação a um único tipo de perturbação do SMD (carga da rede) e é centrada no usuário final, já que o objetivo é maximizar a qualidade e não a utilização dos recursos do sistema.

Geralmente, há dois tipos de adaptação de QoS (Gecsei 1997): adaptação sem realimentação (ou em malha aberta) e adaptação com realimentação (ou em malha fechada). No primeiro tipo, o mecanismo de adaptação solicita aumento ou liberação de largura de banda quando as características do fluxo ou as exigências de QoS mudam. No segundo tipo, o mecanismo de adaptação altera a taxa de bits da aplicação a partir de uma realimentação representada por um ou mais parâmetros de QoS que fornecem uma estimativa da carga corrente da rede. Nesse caso, a informação de uma taxa de perdas baixa leva o mecanismo de adaptação a aumentar a taxa de bits gerada pela aplicação, enquanto uma taxa de perdas elevada conduz a uma diminuição desta[1]. Essa segunda abordagem, que será a focalizada neste trabalho, é a mais comumente utilizada pelos mecanismos de adaptação por ser mais genérica, já que a primeira pressupõe a possibilidade de algum tipo de reserva "on-line" de largura de banda.

4.2.2 Porque Realizar a Adaptação

No projeto de um modelo de QoS, antes da decisão de oferecer mecanismos de adaptação, deve-se analisar se há essa necessidade. Em princípio, em redes onde é possível que a aplicação reserve uma largura de banda compatível com a qualidade a ser oferecida em termos de valores de parâmetros de QoS, a adaptação é desejável, já que em algumas situações a qualidade poderá cair de forma descontrolada e brusca devido a sobrecargas transitórias decorrentes de uma alocação de largura de banda feita de forma otimista, subestimando o pior caso, ou à queda de ligações.

[1] A alteração da taxa de bits da aplicação é a abordagem utilizada pela grande maioria dos mecanismos de adaptação com realimentação. Alguns poucos trabalhos propõem outras estratégias, como o balanceamento de carga através da seleção dinâmica de rotas menos congestionadas (Nahrstedt e Steinmetz 1995).

Em ambientes onde não é possível a reserva de largura de banda (redes melhor-esforço, como a Internet), a adaptação é praticamente obrigatória, caso deseje-se que as aplicações multimídia forneçam uma qualidade medianamente satisfatória. Nesses ambientes, as aplicações são iniciadas independentemente da disponibilidade dos recursos e as sobrecargas são mais freqüentes fazendo com que a qualidade oscile bastante. Em WAN's ("wide area networks") outro fator que conduz à necessidade de adaptação de QoS é a heterogeneidade dos componentes.

A adaptação não implica somente na degradação da QoS: havendo disponibilidade de recursos, ela pode ser disparada para aumentar a qualidade. Há casos, ainda, em que a adaptação não resulta nem em degradação nem em melhora da qualidade. Este é o caso, por exemplo, quando o fluxo de vídeo é comprimido no emissor usando o algoritmo MPEG mas certos receptores só dispõem de decodificadores JPEG. Nesse caso, a ação de adaptação resumir-se-á a uma transcodificação, provavelmente sem ganhos ou perdas em termos da QoS dos parâmetros da camada do usuário.

Um fator secundário que cria a necessidade de adaptação é o melhor aproveitamento (maximização) de largura de banda da rede. Tal fator, contudo, não deve prejudicar o objetivo maior segundo a ótica deste trabalho: a maximização da qualidade oferecida para o usuário final ante o contexto corrente da rede.

4.2.3 O Que é Necessário para a Adaptação

A adaptação de QoS necessita de dois elementos principais: a política de adaptação e o mecanismo de adaptação.

A *política de adaptação* define aspectos como quando, quem e de que forma se dará a adaptação. Uma forma de representação da política de adaptação utiliza, por exemplo, uma série de regras do tipo:

se *taxa de perdas de pacotes é maior que* x então *taxa de bits é* $f(x)$.

No exemplo acima, a política define que quando a variável de realimentação (taxa de perdas de pacotes, no caso) ultrapassar um valor x (uma constante pré-definida), a taxa de bits deverá passar para $f(x)$. A política pode também definir que ações serão realizadas para realizar a adaptação. Por exemplo, a política pode definir que a nova taxa de bits dada por $f(x)$ deve ser obtida através da alteração do fator de quantização do codificador.

O *mecanismo de adaptação* implementa a política, geralmente na forma de um controlador que atuará a partir de informações recebidas do módulo de monitoração (ou simplesmente, monitor). O monitor obtém, de tempos em tempos, os valores dos parâmetros de QoS definidos pela política de adaptação como variáveis de realimentação. O monitor pode ser distribuído nos nós intermediários da rede e/ou nas extremidades (sistemas finais). Algumas variáveis de realimentação usadas por mecanismos de adaptação são: taxa de bits, taxa de perdas de pacotes, atraso, variação do atraso, taxa de ocupação de "buffers" nos nós intermediários da rede, taxa de perdas de quadros e taxa de perdas de "deadlines".

Também, enquanto a política de adaptação define quais parâmetros de QoS serão alterados, o mecanismo de adaptação implementa a forma de atuação do controlador sobre o SMD, *i.e.*, as ações que serão realizadas, de fato, para executar essa alteração. No exemplo citado anteriormente, o mecanismo de adaptação deve dispor dos meios para alterar o fator de quantização visando alcançar a nova taxa de bits dada por $f(x)$. Na Figura 4.1 é mostrado o esquema típico de um mecanismo de adaptação de QoS.

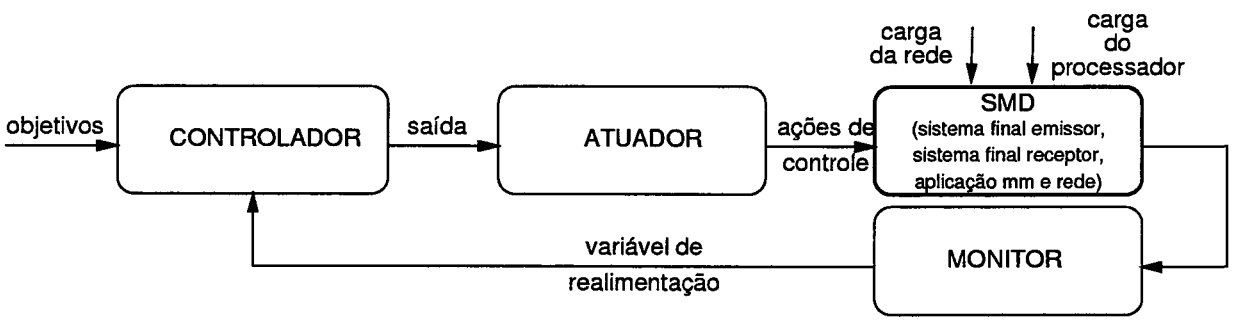

Figura 4.1: Esquema de um mecanismo de adaptação de QoS e do SMD sobre o qual ele atua.

4.2.4 Quando Será Realizada a Adaptação

O momento no qual deve ser iniciado o processo de adaptação ou *ponto de adaptação*[2] ocorre quando a qualidade corrente não pode mais ser mantida ou quando ela pode ser aumentada. Isso implica na possibilidade de existência de dois ou mais pontos de adaptação, expressados na forma de valores absolutos (como no exemplo da Seção 4.2.3, no qual o ponto de adaptação é alcançado quando a taxa de perdas ultrapassa x) ou de intervalos.

[2]Nahrstedt (Nahrstedt et al. 1996) e outros autores chamam esse momento como *ponto de degradação*. Porém, ponto de adaptação parece ser mais adequado já que a adaptação não envolve somente a degradação da QoS.

4.2.5 Onde Ocorrerá a Adaptação

Existem três possibilidades de adaptação (não mutuamente exclusivas) no que concerne ao local aonde ela ocorrerá: sistemas finais emissores, nós intermediários da rede e sistemas finais receptores. O local onde ocorrerá a adaptação é definido pelo *modelo de adaptação*.

A amplitude dos efeitos da adaptação depende diretamente do local onde ela ocorrerá: se a adaptação é realizada nos sistema finais emissores, todas os sistemas finais receptores sentirão a adaptação; se ela ocorre nos nós intermediários da rede, apenas sentirão o efeito da adaptação os sistemas finais que são nós-folhas da sub-árvore desses nós intermediários; por fim, se a adaptação ocorre nos sistemas finais receptores, somente esses sistemas sentirão os efeitos da adaptação. O quanto cada sessão de usuário[3] sofrerá com a adaptação depende da política de adaptação utilizada. Geralmente, quando há custos associados, a prioridade da política de adaptação é manter a maior qualidade possível para aquelas sessões que estão pagando mais. Assim, essas sessões têm maior peso na decisão de como e em quanto adaptar. Em alguns mecanismos, como o de Kawachiaya et al. (Kawachiya et al. 1995), a política de adaptação realiza um balanceamento visando manter a mais alta qualidade possível para todos usuários.

4.2.6 Que Parâmetros Serão Adaptados

Em termos de parâmetros de QoS, o alvo da adaptação à carga da rede é quase sempre a taxa de bits gerada pela aplicação multimídia distribuída. Contudo, para a alteração deste parâmetro de QoS, o mecanismo de adaptação deve alterar o valor de um ou mais parâmetros de QoS da camada da aplicação. Geralmente, apenas um parâmetro da camada de aplicação relacionado à qualidade da imagem, como a freqüência de quadros ou o fator de quantização, tem seu valor mudado visando a alteração da taxa de bits. A preferência pelo uso de parâmetros relacionados à qualidade da imagem sobre parâmetros relacionados à qualidade do som no processo de adaptação decorre de fatores como maior número dos primeiros sobre os últimos, uso de algoritmos hierárquicos para compressão de quadros que, intrinsecamente, oferecem suporte para adaptação, e menor sensibilidade do ser humano à degradação da imagem do que do som. Obviamente, em virtude da interdependência de muitos parâmetros de QoS, a alteração do valor de parâmetros da camada de aplicação implica na alteração dos valores de outros parâmetros, pertencentes ou não à camada de aplicação.

[3] A expressão "sessão de usuário" é usada neste trabalho para designar a sessão da aplicação que é executada em um sistema final receptor.

4.2.7 Como Será Feita a Adaptação

Além de uma política que defina quais parâmetros de QoS deverão ter seus valores alterados, o mecanismo de adaptação de QoS deve dispor de meios para fazer essa alteração. Muitos algoritmos de compressão possuem uma escalabilidade nativa que pode ser explorada diretamente através da alteração de parâmetros usados pelo algoritmo no processo de compressão ou através do uso de ferramentas concebidas para explorar essa escalabilidade. De maneira geral, todas as formas de alteração de valores de parâmetros de QoS acrescentam uma carga adicional ("overhead") que aumenta o atraso fim-a-fim bem como a carga do processador onde a alteração é executada. As formas mais comuns de alteração de parâmetros de QoS são:

1. redução da resolução espacial (Jacobson 1988): de acordo com o estado da rede, a aplicação reduz ou aumenta a resolução espacial. Tal estratégia não pode ser usada com muita freqüência já que ela é percebida pelo usuário final na forma de mudanças no tamanho da janela de exibição;

2. divisão dos fluxos com diferentes qualidades (Delgrossi et al. 1993) (Bolot e Turletti 1998) (Campbell et al. 1998) (Gonçalves et al. 2000): a aplicação gera vários fluxos e a qualidade máxima é obtida através da recepção de todos os fluxos. O número de fluxos que cada sistema final recebe é de acordo com a carga da sub-árvore ao qual ele está associado. A natureza dos fluxos pode ser baseada em diferenças intraquadros, onde cada fluxo representa um nível de resolução (quando o fluxo é comprimido através de um algoritmo de compressão hierárquico) ou interquadros (quando o fluxo é comprimido através de um algoritmo que utiliza compressão temporal);

3. filtragem (Ortega e Ramchandran 1995) (Yeadon et al. 1996) (Eleftheriadis e Anastassiou 1995): o fluxo é submetido a filtros (no sistema final emissor, nos nós intermediários da rede ou nos sistemas finais receptores) que alteram suas características. Existem várias formas de filtragem, dentre as quais: descarte dos coeficientes DCT de freqüência alta em cada bloco de quadros (para algoritmos de compressão baseados em DCT); supressão da cor ou redução da profundidade do pixel e alteração do valor do fator de quantização. Quando a filtragem não é executada em tempo de geração do fluxo, ela pode exigir a descompressão parcial ou total dos quadros;

4. supressão de quadros (Yeadon et al. 1996): a supressão de quadros é usada para redução da freqüência. Os supressores de quadros ("droppers") são um tipo especial de filtro que

decide quais quadros serão eliminados através de uma pesquisa em seus cabeçalhos (que identificam do tipo do quadro, por exemplo, I, P e B, para algoritmos da família MPEG). Um efeito colateral da supressão de quadros, é a provável perda da sincronização labial; e

5. transcodificação (Yeadon et al. 1996): diferentes algoritmos de compressão obtém diferentes taxas de compressão. Existem ferramentas chamadas "video gateways" que permitem a conversão de um fluxo de um formato para outro. Por exemplo, a ferramenta vgw (Amir et al. 1995) converte fluxos de vídeo de/para M-JPEG e H.261. Assim como os filtros, os "video gateways" não localizam-se, necessariamente, nos sistemas finais: eles podem localizar-se em nós intermediários. Um problema da troca de algoritmos de compressão é o antagonismo das características exigências de largura de rede×exigências de processamento: geralmente, quanto menos largura de banda um algoritmo de compressão exige, mais ciclos de processador serão usados pelas tarefas de descompressão. Assim, a troca de um algoritmo de compressão por outro pode implicar no aumento da carga da rede ou dos processadores dos sistemas finais.

4.3 Resumo e Discussão

Um mecanismo de adaptação de QoS para aplicações multimídia é necessário ainda que seja possível a reserva de recursos, já que mesmo nessa situação existe a possibilidade de ocorrência de insuficiência (ou aumento de disponibilidade) de largura de banda.

A forma mais comum de adaptação é aquela baseada em malha fechada. Uma variável de realimentação é usada pelos mecanismos de adaptação de QoS para estimar a carga da rede.

O processo de adaptação de QoS deve culminar com a alteração dos valores de um ou mais parâmetros da camada da aplicação. A alteração da taxa de bits da aplicação frente à mudança de disponibilidade de largura de banda pode ser feita, por exemplo, através das mudanças da freqüência de quadros e/ou profundidade do pixel e/ou freqüência de amostras de áudio etc.

Devido ao fato da qualidade da imagem ser o maior consumidor de recursos, a maior parte dos mecanismos de adaptação concentra suas ações sobre os parâmetros de QoS relacionados à imagem. Contudo, o número de parâmetros candidatos à adaptação nas propostas atualmente existentes limita-se, geralmente, a um único parâmetro, desconsiderando que o usuário percebe a qualidade como um todo.

Alguns problemas comuns a várias propostas relacionadas à adaptação de QoS são relacionados à política de adaptação, que estima o estado do SMD através de intervalos de valores da variável de realimentação, negligenciando a incerteza inerente a essa determinação, e que determina que apenas um parâmetro de QoS será alterado no processo; e ao mecanismo de adaptação, que é ligado a alguma tecnologia de rede específica.

Devido aos problemas acima mencionados, neste trabalho é proposta uma forma de adaptação multidimensional (envolvendo vários parâmetros de QoS), independente de plataformas e tecnologias, baseada no uso de lógica nebulosa, uma abordagem concebida para tratar exatamente a questão da incerteza. Tal proposta será descrita no próximo capítulo.

Capítulo 5

ADAPTAÇÃO DE QoS BASEADA EM CONTROLE NEBULOSO

5.1 Introdução

No capítulo anterior foi visto que a adaptação de QoS é realizada através de mecanismos baseados em políticas de adaptação. As políticas geralmente utilizam uma variável fornecida por realimentação para estimar o estado do SMD. Também foi visto que o processo de adaptação, usualmente, culmina com a alteração de algum parâmetro da camada de aplicação. Essa alteração é refletida em uma mudança na taxa de bits da aplicação, quando a adaptação ocorre como uma resposta à carga da rede.

Neste capítulo serão descritos alguns dos problemas encontrados na abordagem acima e será proposta uma solução para tais problemas através do uso de um mecanismo de adaptação de QoS baseado em controle nebuloso utilizando o modelo clássico e o modelo de interpolação.

5.2 Trabalhos Relacionados

Nesta seção, são apresentados alguns trabalhos relacionados à adaptação de QoS. Tal levantamento não pretende ser exaustivo, mas apenas representativo de algumas abordagens baseadas em adaptação com realimentação.

A maior parte dos mecanismos de adaptação propostos na literatura é direcionada para aplicações multimídia distribuídas cuja plataforma-alvo é a Internet, por esse ambiente ser totalmente melhor-esforço, o que torna a adaptação da QoS absolutamente necessária para que a

qualidade das aplicações multimídia mantenha-se em um nível minimamente satisfatório. Deve-se salientar, contudo, que muitas das propostas apresentadas podem ser utilizadas também em redes que ofereçam a possibilidade de reserva de recursos, como é o caso da rede ATM.

Além de serem direcionados para Internet, muitos mecanismos de adaptação de QoS fazem uso de protocolos de comunicação da camada da aplicação que oferecem facilidades para monitoramento de QoS. Dentre esses protocolos, sem dúvida o mais utilizado é o RTP. O RTP ("Real-Time Transport Protocol") (Schulzrinne et al. 1996) é um protocolo concebido para atuar como uma interface entre aplicações de tempo real e protocolos da camada de transporte. O RTP dispõe de um protocolo do controle chamado RTCP ("Real-Time Control Protocol") que pode ser usado para a obtenção do valor de uma série de parâmetros de QoS não-controláveis. Uma mensagem RTCP consiste de um número de pacotes, cada um com seu próprio código de tipo e indicação de tamanho, tendo um formato bastante similar aos pacotes de dados. As mensagens RTCP são enviadas periodicamente de forma multiponto para o mesmo grupo multiponto que recebe os pacotes de dados. Assim, elas servem também para indicar quais membros ainda fazem parte da sessão, mesmo na ausência de emissão de dados. As funções principais do RTCP são fornecidas pelos pacotes RR ("receiver report"), enviados pelos sistemas finais receptores para os sistemas finais emissores, e SR ("sender report"), enviados dos sistemas finais emissores para os sistemas finais receptores. Os pacotes RR contêm o número de seqüência mais elevado recebido, o número de pacotes perdidos, a variação do atraso e os registros de tempo ("timestamps") necessários para estimar o atraso total ("round-trip delay") entre o sistema final emissor e o sistema final receptor que enviou o pacote RR; os pacotes SR contêm informações que permitem estimar a taxa de transmissão real gerada pela aplicação.

A seguir, serão descritos três mecanismos de adaptação de QoS. Todos eles são posicionados nos sistema final emissor e são realimentados pela taxa de perdas de pacotes, obtida através dos pacotes RR do protocolo RTCP.

5.2.1 Mecanismo de Controle de Aplicação Fim-a-fim

Em (Busse et al. 1995) é proposto um mecanismo de adaptação de QoS referenciado como *Mecanismo de Controle de Aplicação Fim-a-fim* ("End-to-end Application Control Mechanism" - EACM). O mecanismo é realimentado por todos os sistemas finais receptores com as taxas de perdas de pacotes. Baseado nesta métrica, é determinado o estado da rede, conforme ele é visto pelos sistemas finais receptores, e a taxa de bits é ajustada por um regulador linear com uma

zona morta ("dead zone"). A taxa da perdas é usada como um indicador de congestionamento. Um filtro passa baixa é usado para suavizar as oscilações das perdas. A taxa suavizada de perdas do $i^{\text{ésimo}}$ sistema final receptor ($perdas_i^f$) é calculada de acordo com a equação abaixo:

$$perdas_i^f(t_n) = (1 - \beta) \times perdas_i(t_{n-1}) + \beta \times perdas_i(t_n) \tag{5.1}$$

onde β é uma constante ($0 \leq \beta \leq 1$), $perdas_i(t_{n-1})$ é a taxa de perdas do $i^{\text{ésimo}}$ sistema final receptor no tempo t_{n-1}, e $perdas_i(t_n)$ é a taxa de perdas do $i^{\text{ésimo}}$ sistema final receptor no tempo t_n. Quanto menor o valor de β, maior a influência de $perdas_i(t_n)$; quanto maior o valor de β, maior a influência de $perdas_i(t_{n-1})$.

O valor de $perdas_i^f$ é usado pelo mecanismo para determinar o estado como a rede é vista pelo $i^{\text{ésimo}}$ sistema final receptor: DESCARREGADA, CARREGADA ou CONGESTIONADA. Tais estados representam, de fato, três intervalos: $[0, perdas_u[$, $[perdas_u, perdas_l[$ e $[perdas_l, 1]$, respectivamente. O limite $perdas_u$ deve ser escolhido de modo que o número de pacotes perdidos seja ainda aceitável. O segundo intervalo deve ser grande suficiente para evitar oscilações de QoS. A escolha dos limites é arbitrária e tem que ser justificada através de resultados experimentais.

A decisão do ajuste é feita através do exame da proporção de sistemas finais receptores descarregados, carregados e congestionados. Essas proporções são comparadas com dois pontos-limite cujos valores são definidos arbitrariamente para decidir a ação de controle. A Figura 5.1 mostra o algoritmo do EACM onde N é o número total de sistemas finais receptores, N_c é o número de sistemas finais receptores no estado congestionado e N_l é o número de sistemas finais receptores no estado carregado; N_d e N_h são os dois pontos-limite. Bps é a taxa de bits gerada pela aplicação e calculada pelo controlador, Bps_R é a taxa de bits real (determinada a partir dos pacotes SR) que inclui informações introduzidas pela pilha de protocolos e parte do fluxo RTCP, e Bps_{min} e Bps_{max} são as taxas de bits mínima e máxima permitidas; ν é um fator multiplicador (entre 0 e 1) e η é um fator aditivo representando uma taxa de bits; ambas constantes são arbitrárias.

5.2.2 Algoritmo de Ajuste Direto

Em (Sisalem 1998) é descrito um mecanismo semelhante ao anterior, também baseado no uso do RTP/RTCP e de funções lineares, chamado *Algoritmo de Ajuste Direto* ("Direct Adjustment

se $\left(\frac{N_c}{N}\right) \leq N_d$ então $d \leftarrow DECREMENTA$
senão se $\frac{N_l}{N} \leq N_h$ então $d \leftarrow MANTEM$
senão $d \leftarrow INCREMENTA$
case d
$DECREMENTA$: $Bps \leftarrow max(Bps_R \times \nu, Bps_{min})$
$INCREMENTA$: $Bps \leftarrow min(Bps_R + \eta, Bps_{max})$

Figura 5.1: Algoritmo do EACM.

Algorithm" - DAA). Nesse mecanismo, o sistema final emissor inicia a sessão no estado descarregado, emitindo um fluxo com uma taxa de bits pré-definida e aumenta essa taxa com um fator aditivo η dividido pelo mínimo entre o número total de membros da sessão (N) e um limiar de escalamento (th_{scale}):

$$Bps \leftarrow Bps_R + \frac{\eta}{min(N, th_{scale})} \tag{5.2}$$

onde th_{scale} é um fator que calcula o número máximo de membros da sessão considerando a largura de banda disponível para o tráfego RTCP. Ele é obtido através da equação:

$$th_{scale} = \frac{I_{min} \times Bps_{RTCP}}{S_{RTCP}} \tag{5.3}$$

I_{min} é o intervalo mínimo entre dois pacotes RTCP (ajustado para 5 segundos), Bps_{RTCP} é a largura de banda para o tráfego RTCP (geralmente 5% do tráfego de dados), e S_{RTCP} é o tamanho do pacote RTCP. Quando um sistema final receptor i informa uma taxa da perdas (filtrada como no mecanismo de Busse) maior do que o limite pré-definido ($perdas_c$), o sistema final emissor entra no estado congestionado e a taxa de bits é reduzida na seguinte proporção:

$$Bps \leftarrow Bps_R \times (1 - perdas_i^f + perdas_c) \tag{5.4}$$

A identidade do sistema final receptor é guardada em *MembroPerdedor* e o valor de perdas reportado ($perdas_i^f$) é salvo em $MaisAltaPerda$. Mensagens com valores de perdas menores do que $MaisAltaPerda$ são ignoradas; se um sistema final receptor j ($j \neq i$) reporta valores de perdas maiores do que $MaisAltaPerda$, a taxa de bits sofre uma redução adicional de acordo com a equação abaixo:

$$Bps \leftarrow Bps_R \times (1 - perdas_j^f + MaisAltaPerda) \tag{5.5}$$

MembroPerdedor recebe então a identidade do sistema final receptor j e $MaisAltaPerda$ recebe $perdas_j^f$. Após receber uma mensagem de *MembroPerdedor* com valores de perdas menores do que $perdas_c$, o sistema final emissor volta para o estado descarregado. Como em (Busse et al. 1995), pode-se especificar as taxas de bits mínima Bps_{min} e máxima Bps_{max} permitidas.

5.2.3 Mecanismo de Adaptação para a WWW

Em (Fry et al. 1996), é descrito um mecanismo de adaptação de QoS direcionado para aplicações da "World Wide Web" (WWW) que visa solucionar um dos problemas relacionados à execução de aplicações multimídia sobre a Internet: a dificuldade da reprodução de dados de mídia contínuas em tempo real, já que a tecnologia corrente de navegadores exige que os dados sejam integralmente carregados na memória da estação do cliente para, só então, serem exibidos. Além do "overhead", o cliente, eventualmente, sequer terá memória suficiente para carregar as imagens.

O mecanismo de adaptação é composto de um servidor e um cliente de mídias contínuas (MC), um do lado do servidor HTTP e outro do lado do cliente (o navegador). O servidor MC é controlado pelo usuário WWW via HTTP usando uma CGI ("Common Gateway Interface"), usada também para a negociação dos valores de QoS. Existe uma conexão adicional entre o servidor e o cliente de MC por onde trafegam os dados das mídias e os dados para realimentação do módulo de adaptação de QoS. A descrição dos limites e prioridades de degradação é feita através do caminho de degradação, já descrito no Capítulo 3. O caminho de degradação é representado por uma tabela onde cada entrada contém a resolução e a freqüência dos quadros bem como o algoritmo de compressão a ser usado. Antes do início da sessão da aplicação propriamente dita, o usuário fornece suas preferências em relação a cada um desses parâmetros, atribuindo pesos que expressam as prioridades de degradação. Ele também informa se dispõe ou não de placa de decodificação. A cada nível de QoS, é associada a largura de banda necessária.

Quando a largura de banda disponível não permite a manutenção do nível de QoS corrente, o que é detectado quando a taxa de perdas de pacotes (novamente determinada através dos pacotes RR) ultrapassa 10%, o mecanismo de adaptação de QoS é disparado, passando para o nível de QoS imediatamente abaixo do corrente, determinado pelo caminho de degradação. O cliente de MC envia um pacote RR por segundo para que o servidor de MC do lado do servidor calcule a taxa de perdas de pacotes e realize a adaptação (se necessário).

5.2.4 Limitações

De maneira geral, a maior parte dos mecanismos de adaptação de QoS propostos na literatura não contemplam a *considerável incerteza presente na determinação desse estado.* Tal incerteza é decorrente dos seguinte aspectos:

1. a natureza das variáveis de realimentação (independentemente de quais sejam), que fornece apenas uma idéia vaga da carga da rede. A taxa de perdas de pacotes, por exemplo, apenas indica a presença de carga, sem precisar suas características e seus efeitos sobre a qualidade;

2. o valor da variável de realimentação, que torna-se rapidamente desatualizado em virtude da dinâmica do SMD, especialmente em WAN's, nas quais usuários entram e saem com grande freqüência, aplicações emitem rajadas ("burst") de pacotes de tempos em tempos, conexões caem etc.; e

3. o próprio conceito de "estado da rede", intrinsecamente vago.

Além de não contemplarem a incerteza presente na determinação do estado da rede, os mecanismos de adaptação de QoS em geral apresentam um ou mais dos seguintes problemas:

1. eles são associados a determinadas tecnologias de rede (ATM, especialmente);

2. eles não são centrados no usuário final, já que avaliam a qualidade utilizando apenas uma dimensão, que muitas vezes nem sequer é representada por parâmetros da camada de aplicação; e

3. eles não mostram quais (e como) parâmetros de QoS da camada de aplicação terão seus valores alterados para que a taxa de bits calculada seja realmente alcançada.

Em virtude dos problemas acima expostos, neste trabalho é proposto um mecanismo de adaptação de QoS baseado no uso de controle nebuloso, uma abordagem orientada exatamente para tratar a questão da incerteza, e na função grau de qualidade, vista no Capítulo 3.

5.3 Justificativa

SMD's são sistemas cujo comportamento é extremamente difícil de ser modelado, em decorrência da existência de variáveis (particularmente, taxa de perdas de pacotes e atraso)

que sofrem mudanças de valor de maneira imprevisível. Em virtude do uso de algoritmos de compressão, o reflexo dessas variáveis sobre outras também é, em muitos casos, desconhecido ou impossível de ser matematicamente formulado, como é o caso do efeito da taxa de perdas de pacotes sobre as taxas de perdas de quadros de vídeo ou amostras de áudio. Tais dificuldades tornam os SMD's sistemas difíceis de serem modelados. Essa característica, *per si*, torna adequado o uso de controle nebuloso para adaptação de QoS, já que essa abordagem dispensa o uso de modelos analíticos complexos e permite que o controlador seja refinado a partir da experiência obtida no decorrer do tempo. Além disso, um mecanismo de adaptação de QoS baseado no uso de controle nebuloso é adequado já que, segundo (Correa 1999), "em geral, controladores nebulosos encontram maior utilidade em sistemas não-lineares, sendo capazes de suportar muito bem perturbações e plantas com altos níveis de ruídos", o que combina com as características de SMD's, nos quais a qualidade de apresentação é não-linear em relação aos parâmetros de QoS, podendo apenas ser empiricamente avaliada: a duplicação da freqüência de quadros, por exemplo, não significa que o usuário irá considerar a qualidade duas vezes melhor. No caso específico deste trabalho, a adequação do uso de controle nebuloso é reforçada pelo fato do critério de desempenho do mecanismo - a qualidade da apresentação - ser intrinsecamente vago.

Apesar das várias características expostas acima que corroboram com a adequação do uso de controle nebuloso para adaptação de QoS, há ainda poucos trabalhos que contemplam essa possibilidade, demonstrando sua viabilidade e confirmando essa adequação através de resultados.

Em (Bogatinovski et al. 1998), é apresentado um controlador nebuloso para regular o tráfego de aplicações de videoconferência executadas sobre redes ATM usando o serviço ABR. O controlador é usado para calcular um sinal de controle que é, por sua vez, usado no cálculo da taxa explícita ("explicit rate" ou ER) que é enviada para os sistemas finais emissores via células RM (a ER define a taxa de bits máxima dos sistemas finais emissores). Esse controlador difere do proposto neste trabalho em vários aspectos: ele é amarrado à tecnologia ATM, o valor da variável de realimentação é calculado nos "switchs" a partir da taxa de ocupação dos "buffers" e ele altera apenas o fator de quantização para que a taxa de bits calculada seja alcançada. A idéia do cálculo da variável de realimentação nos nós intermediários da rede, entretanto, é interessante como uma alternativa para diminuir a possibilidade de explosão de realimentação.

Em (Li e Nahrstedt 1999), é proposto o uso da Teoria de Conjuntos Nebulosos para traduzir pedidos de recursos para parâmetros de ajuste. O modelo da adaptação usado é muito diferente do modelo apresentado no presente trabalho em termos de variáveis e regras usadas pelo controlador nebuloso, mas muitos aspectos teóricos apresentados nele podem ser usados como um auxílio para uma melhor formalização de abordagem aqui proposta.

5.4 Mecanismo de Adaptação de QoS Nebuloso

O mecanismo de adaptação de QoS aqui proposto, assim como a maior parte de seus congêneres, é posicionado no sistema final emissor e recebe informações de realimentação dos sistemas finais receptores.

As principais diferenças deste mecanismo em relação a outros propostos na literatura são:

1. ele não é associado a nenhuma tecnologia de rede, algoritmo de compressão ou protocolo de comunicação específicos;

2. a maximização do uso da largura de banda é uma conseqüência e não um fim, já que seu objetivo é a maximização da qualidade oferecida aos usuários finais;

3. ele faz uso de um controlador nebuloso (CN) para calcular a nova taxa de bits, o que faz com que esse cálculo implicitamente considere a incerteza presente na determinação do estado do SMD; e

4. com o intuito de moldar a taxa de bits gerada pela aplicação multimídia àquela calculada pelo CN, ele atua sobre vários parâmetros de QoS ao invés de um único, fazendo com que as mudanças de qualidade percebidas pelo usuário final não sejam tão drásticas.

O mecanismo pode ser usado tanto em redes melhor-esforço quanto em redes que oferecem a possibilidade de reserva de largura de banda. Seu objetivo genérico é oferecer a maior qualidade possível para a maior parte dos sistemas finais receptores (usuários finais). O mecanismo de adaptação de QoS nebuloso é mostrado na Figura 5.2. Um monitor observa os valores de determinadas variáveis do SMD que devem fornecer uma idéia da qualidade que cada usuário está recebendo. Com essas informações, ele determina o valor da variável de realimentação, utilizada pelo CN. Para cada conjunto nebuloso associado à variável de realimentação, o CN obtém a compatibilidade do valor atual medido para essa variável em relação às funções de pertinência

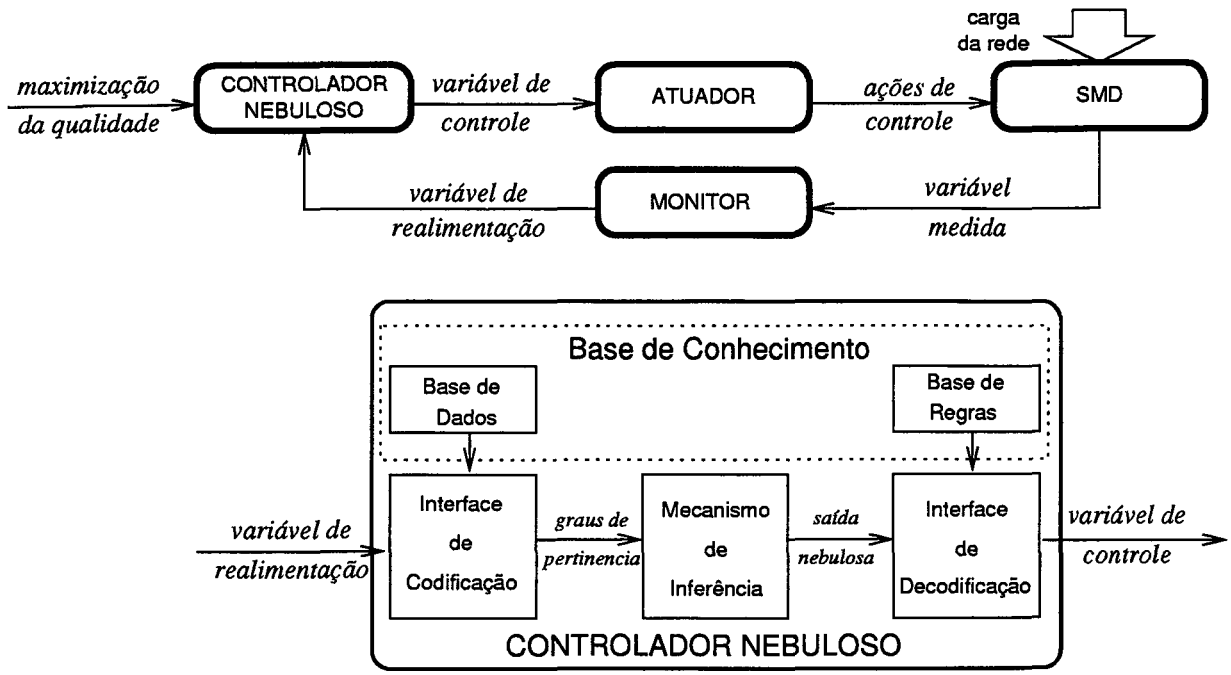

Figura 5.2: Diagrama de blocos do mecanismo de controle nebuloso.

contidas na base de dados, em um processo conhecido como *codificação* (vide Apêndice B). O mecanismo de inferência usa esses graus de pertinência e as regras que estabelecem a política de adaptação para calcular a saída do CN, na forma de variáveis de controle. Conforme o modelo de CN usado, a saída deve ser decodificada ("defuzzified"). Essa saída deve ser uma informação que possa ser mapeada para um nível de QoS. Exemplos de saída possíveis são o grau de qualidade e a taxa de bits. O atuador mapea a saída do CN para um nível de QoS e realiza as ações de controle necessárias para alterar os valores do parâmetros de QoS do fluxo de acordo com o nível de QoS obtido.

5.4.1 Framework

O mecanismo de adaptação de QoS nebuloso proposto neste trabalho faz uso de uma tabela contendo os níveis de QoS específicos para um dado codificador, com os graus de qualidade e taxa de bits associados. A construção dessa tabela - referenciada como $\Omega_{NiveisQoS}$ - é um processo composto de uma série de etapas, realizadas quando da construção do mecanismo de adaptação de QoS (portanto, sem a participação do usuário final da aplicação multimídia). As etapas necessárias para a construção de $\Omega_{NiveisQoS}$ são mostradas na Figura 5.3.

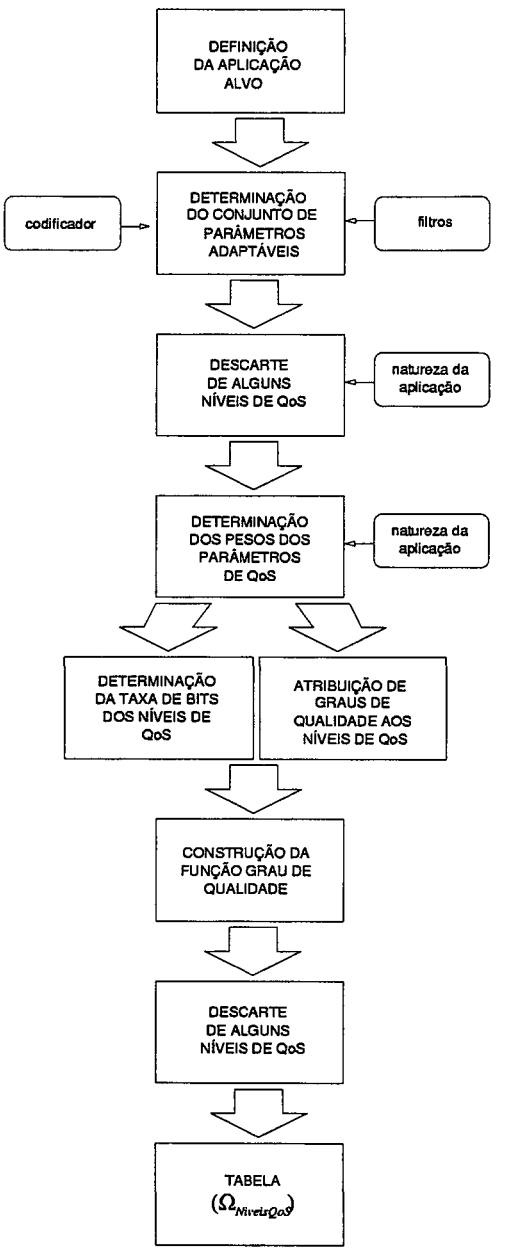

Figura 5.3: "Framework"com os passos que antecedem à adaptação de QoS.

O primeiro passo consiste na definição da aplicação-alvo. Apesar do mecanismo ser genérico, sua política de adaptação utilizará informações que estão intimamente relacionadas à natureza da aplicação (particularmente, a importância relativa de cada dimensão ou parâmetro de QoS na composição da qualidade) e que estarão representadas em $\Omega_{NiveisQoS}$.

Outra informação importante representada em $\Omega_{NiveisQoS}$ é a qualidade de cada nível de QoS, representada através do grau de qualidade visto no Capítulo 3. Assim, é necessária a obtenção da função grau de qualidade ($\mathcal{QoS}$), que deve ter como argumentos apenas aqueles parâmetros da camada de aplicação que poderão ter seus valores alterados pelo mecanismo de adaptação durante a sessão da aplicação. Esse conjunto de parâmetros tem forte relação com o algoritmo de compressão e com o codificador utilizados, bem como com os mecanismos de filtragem eventualmente disponíveis. Assim, o segundo passo para construção de $\Omega_{NiveisQoS}$ consiste na obtenção do conjunto de parâmetros dinamicamente configuráveis, ou

$$\{\rho_1, \rho_2, ..., \rho_m\} \in \{\rho_1, \rho_2, ..., \rho_n\}\ m \leq n$$

Esse conjunto determina os níveis de QoS que poderão ser usados pelo mecanismo de adaptação.

Determinado o conjunto de parâmetros de QoS (e os domínios associados a cada um deles), tem-se o conjunto contendo todos os níveis de QoS. Como a natureza da aplicação define uma série de restrições em relação a valores de parâmetros de QoS que deve ser considerada quando da adaptação (vide Capítulo 2), muitos níveis de QoS que não satisfazem tais restrições deverão ser descartados. Aos níveis de QoS restantes (que satisfazem as restrições) serão atribuídos graus de qualidade obtidos a partir de entrevistas com grupos de usuários utilizando uma das abordagens descritas no Capítulo 2. Os clipes utilizados para a avaliação deverão ser escolhidos em conformidade com o tipo de vídeo da aplicação-alvo. Por exemplo, se a aplicação-alvo consiste em uma videoconferência do tipo seminário, os clipes poderão ser do tipo "cabeças falantes".

Para ser utilizada como elemento norteador da adaptação de QoS, a função $\mathcal{QoS}$ necessita considerar a natureza da aplicação-alvo. No "framework" aqui proposto, isso é feito através do arbitramento de pesos para cada um dos parâmetros de QoS do conjunto acima, ou seja, na definição de w_{ρ_1}, w_{ρ_2}, ...,w_{ρ_m} ($\sum_{i=1}^{m} w_{\rho_i} = 1$). Esses pesos, diferentemente daqueles vistos no Capítulo 3, são atribuídos intuitivamente conforme a natureza da aplicação, refletindo a importância relativa de ρ_i para determinado tipo de aplicação e não na composição da qualidade.

Através do uso desses pesos, o grau de qualidade considera não só as preferências do usuário mas também a natureza da aplicação. Neste caso, para o nível de QoS

$$L_j =< \rho_{1_j}, \rho_{2_j}, ..., \rho_{m_j} >,$$

o grau de qualidade é obtido a partir da equação abaixo:

$$\mathcal{QoS}(< \rho_{1_j}, \rho_{2_j}, ..., \rho_{m_j} >) = min(\upsilon_{\rho_1}(\rho_{1_j}) \times \omega_{\rho_1} \times w_{\rho_1}, \upsilon_{\rho_2}(\rho_{2_j}) \times \omega_{\rho_2} \times w_{\rho_2}, ..., \upsilon_{\rho_m}(\rho_{m_j}) \times \omega_{\rho_n} \times w_{\rho_m}), \tag{5.6}$$

onde

υ_{ρ_i} : função utilidade obtida para o parâmetro ρ_i

ω_{ρ_i} : importância relativa do parâmetro ρ_i para o usuário final

w_{ρ_i} : importância relativa do parâmetro ρ_i para a aplicação-alvo

Paralelamente, são determinadas as taxas de bits associadas aos níveis de QoS. Essas taxas deverão ser obtidas através de medições sobre clipes com os diferentes níveis de QoS. Para cada nível de QoS, será considerado o pior caso. Tais taxas têm que ser obtidas empiricamente, representando o pior caso. Isso pode ser realizado medindo-se a taxa de bits de diversos clipes abrangendo todos os níveis de QoS possíveis, com muito movimento e contendo apenas quadros do tipo I (o que reduz bastante a taxa de compressão). A combinação da taxa de bits/níveis de QoS/grau de qualidade é representada através de uma tabela referenciada como $\Omega_{NiveisQoS}$. As medições podem ser estendidas para outras dimensões de custo (em termos de consumo de recursos), como memória e ciclos de processador. Contudo, dentro do escopo deste trabalho, tais dimensões adicionais são desnecessárias.

A tabela $\Omega_{NiveisQoS}$ tem, então, $\prod_{i=1}^{n} \mathcal{C}(\rho_i)$ $(m+2)$-tuplas do tipo

$$< Bps, \rho_1, \rho_2, ..., \rho_n, Q\hat{o}S >$$

onde $\mathcal{C}(\rho_i)$ é cardinalidade do conjunto Ω_{ρ_i} e Bps é a taxa de bits associada ao nível de QoS contido na tupla.

Para diminuir o número de entradas da tabela $\Omega_{NiveisQoS}$, pode-se descartar todos níveis de QoS L_j pertencentes ao conjunto abaixo:

$$\{L_j | \exists L_i, (Q\hat{o}S_j \leq Q\hat{o}S_i) \wedge (Bps_j \geq Bps_i)\} \tag{5.7}$$

onde

L_k : um nível de QoS qualquer

$Q\hat{o}S_k$: o grau de qualidade do nível de QoS L_k

Bps_k : a taxa de bits associada ao nível de QoS L_k

Isso significa que, dado um nível de QoS qualquer, serão descartados todos aqueles com um grau de qualidade menor ou igual ao dele mas que exijam maior largura de banda. Assim, $\Omega_{NiveisQoS}$ irá conter apenas os níveis de QoS mais "econômicos". De fato, existirão níveis de QoS com diferentes graus de qualidade mas com as mesmas necessidades em termos de largura de banda e vice-versa, o que derruba uma falsa premissa que muitos autores assumem em relação à qualidade, ou seja, que ela é diretamente proporcional à taxa de bits. Apesar de uma largura de banda maior *possibilitar* uma qualidade superior, a qualidade *não é necessariamente* proporcional à taxa de bits. Isso decorre do fato de alguns parâmetros de QoS terem um grande impacto na percepção do usuário em relação à qualidade mas não em relação às necessidades de largura de banda. Em contrapartida, há outros que têm pouco impacto na qualidade percebida mas grande impacto nas necessidades de largura de banda, já que a qualidade é percebida pelo usuário considerando-se *todos* os parâmetros de QoS da camada de aplicação.

Outra medida para redução de $\Omega_{NiveisQoS}$ consiste no estabelecimento de um grau de qualidade limite para degradação (0.5, por exemplo), descartando-se também todos os níveis de QoS cujo grau qualidade é inferior ao limite.

A seguir, são descritas duas possíveis instanciações do mecanismo de adaptação de QoS nebuloso. A diferença entre elas refere-se ao modelo de adaptação de QoS, às variáveis de realimentação e ao paradigma de controle nebuloso utilizados. A primeira abordagem para adaptação de QoS baseada em controle nebuloso será referenciada como *mecanismo de adaptação com controle da taxa de bits* (Koliver et al. 2000 b) (Koliver e Farines 2001); a segunda abordagem será referenciada como *mecanismo de adaptação com controle do grau de qualidade* (Koliver et al. 2001). Ambas abordagens têm como objetivo maximizar o grau de qualidade enviado pela aplicação diante do contexto corrente do SMD.

Visando facilitar o entendimento, o modelo de adaptação de QoS usado nas abordagens considera que a aplicação multimídia distribuída é do tipo 1:N, isto é, um sistema final emissor gera um fluxo multimídia que é distribuído para N sistemas finais receptores.

5.4.2 Mecanismo de Adaptação com Controle da Taxa de Bits

Nesta seção, é descrita a primeira forma de mecanismo da adaptação de QoS proposta neste trabalho. De acordo com o modelo de adaptação usado (Figura 5.4), a aplicação envia o fluxo com uma taxa de bits Bps cujo valor inicial é Bps_{max}. Essa taxa aumenta ao entrar na rede devido às informações introduzidas pela pilha de protocolos. A taxa realmente transmitida é a taxa de bits real $Bps_R > Bps$. O $i^{ésimo}$ sistema final receptor ($i = 1, 2, ..., N$) recebe o fluxo com uma taxa de bits de recepção Bps_{r_i} tal que $Bps_{r_i} \leq Bps_R$ em virtude das perdas sofridas na rede. As taxas de perdas de pacotes $perdas_i$ dos sistemas finais receptores são enviadas para o sistema final emissor.

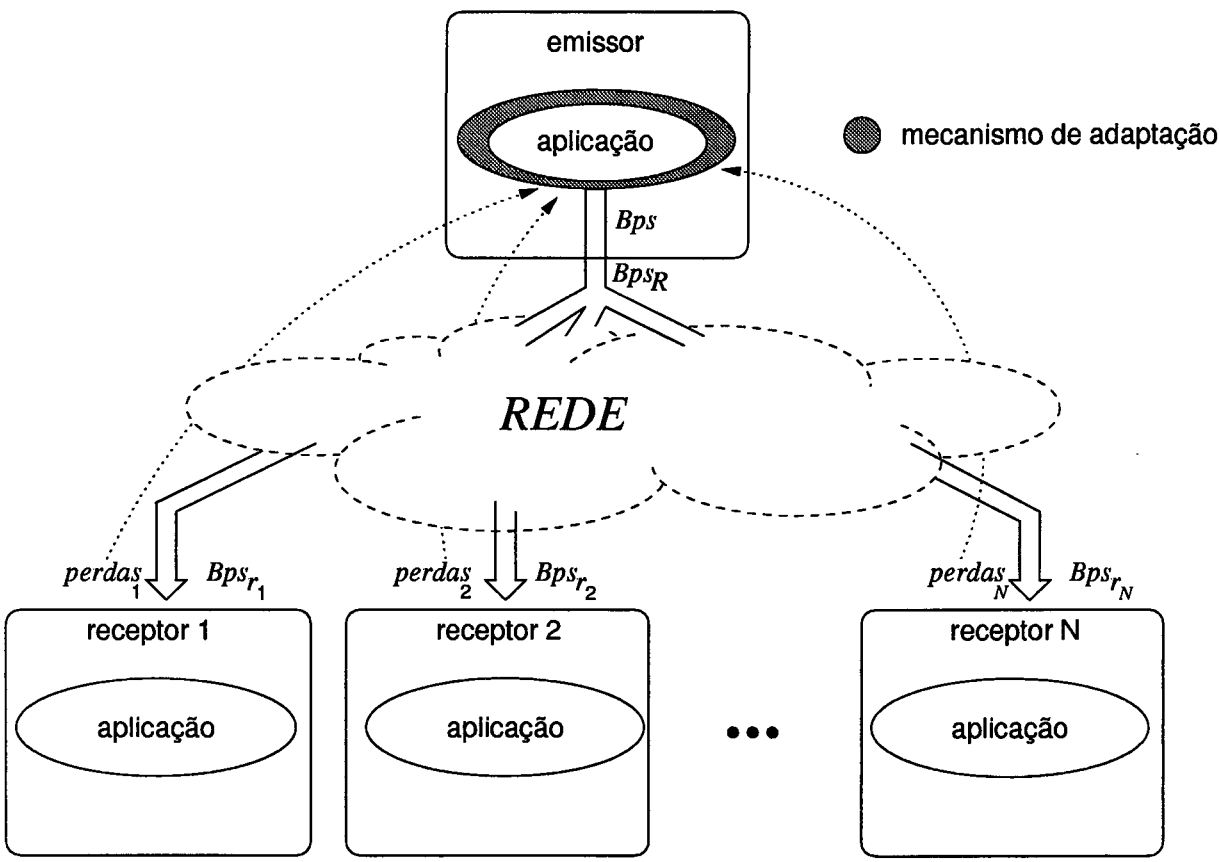

Figura 5.4: Modelo de adaptação de QoS.

A Figura 5.5 representa o esquema de controle. A perturbação tratada pelo mecanismo de adaptação é a carga da rede. As variáveis de realimentação usadas, coletadas pelo monitor, são a taxa de bits real e as taxas de perdas de pacotes de cada sistema final receptor. Considerando-se que a aplicação envia o fluxo multimídia através do protocolo RTP, os valores dessas variáveis podem ser obtidos através do protocolo de controle RTCP. A variável controlada pelo CN é a

taxa de bits, calculada a partir da taxa de perdas agregada. O atuador mapeia essa taxa para um nível de QoS e realiza as ações de controle sobre a aplicação multimídia para que o fluxo passe a ser emitido com esse nível de QoS obtido. Todos elementos do mecanismo de adaptação (CN, monitor e atuador) localizam-se no sistema final emissor.

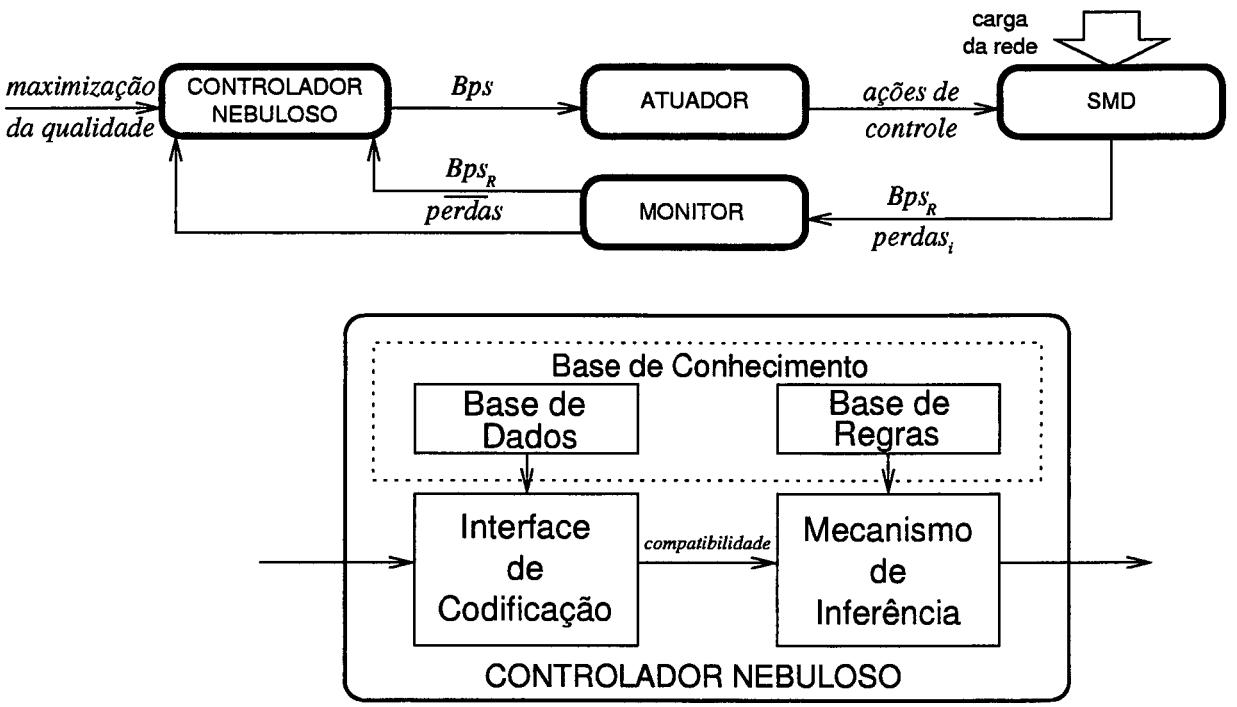

Figura 5.5: Esquema de controle.

A seguir, serão detalhados os módulos do esquema de controle.

5.4.2.1 Monitor

O monitor obtém a taxa de perdas de pacotes de cada sistema final receptor ($perdas_i$, $i = 1, 2, ..., N$) através dos pacotes RR mais recentes. Visando suavizar as oscilações de curta duração, as taxas de perdas passam por um filtro passa baixa. A taxa suavizada de perdas do $i^{\text{ésimo}}$-sistema final receptor ($perdas_i^f$) é calculada como nos mecanismos EACM e DAA, vistos na Seção5.2.

As taxas de perdas suavizadas são usadas para calcular a *taxa de perdas de pacotes agregada* ($\overline{perdas}$). Diferentes políticas - referenciadas neste trabalho como *políticas de agregação da variável de realimentação* (ou simplesmente, políticas de agregação) - podem ser usadas para o cálculo de $\overline{perdas}$. Alguns exemplos de políticas de agregação são a média aritmética, a média

harmônica, a média aritmética ou harmônica ponderada[1] e o pior caso.

Um problema comum a todas essas políticas é a justiça quando da adaptação. Tal problema ocorre principalmente quando os membros da sessão recebem dados através de ligações com capacidades muito diferentes, especialmente em WAN'S. De acordo com (Sisalem 1998), este problema - conhecido como *"beat down problem"* - ocorre em conseqüência do aumento da probabilidade de perdas a cada ligação adicional, fazendo com que sistemas finais receptores distantes reportem mensagens de realimentação com valores de perdas muito mais elevados do que aquelas fornecidas pelos sistemas finais receptores mais próximos. Isso possibilita que, independentemente da política utilizada, certos sistemas finais receptores recebam uma qualidade aquém ou além de suas possibilidades, tanto em termos de disponibilidade de largura de banda quanto poder de processamento. Para serviços pagos, onde é possível a reserva de recursos, a política de agregação proposta neste trabalho é a do pior caso, já que tal política faz com que o processo de adaptação garanta uma qualidade mínima para todos os sistemas finais receptores. Se o ambiente é melhor-esforço, a política de agregação pode usar a média geométrica, ou seja,

$$\overline{perdas} = \sqrt[n]{perdas_1^f \times perdas_2^f \times ... \times perdas_n^f}. \quad (5.8)$$

A escolha de uma política de agregação baseada na média geométrica deve-se ao fato desta média ser a menos influenciada por valores extremos, ideal para um ambiente melhor-esforço, no qual a política de adaptação deve tentar privilegiar a maioria dos usuários.

Além de $perdas_i$, o monitor obtém, através dos pacotes SR mais recentes, a taxa de bits real Bps_R. Estas variáveis de realimentação são fornecidas para o CN.

5.4.2.2 Controlador Nebuloso

O CN, posicionado no sistema final emissor, segue o modelo de CN's de Takagi-Sugeno, no qual as saídas das regras (Figura 5.6) são combinações ou funções lineares das entradas. De fato, cada regra representa um controlador linear e o CN faz a interpolação entre elas (vide Apêndice B).

De acordo com as regras acima, a taxa de bits é decrementada quando as rede está congestionada, mantida constante, quando a rede está carregada, ou incrementada quando a rede está descarregada.

[1] Uma alternativa para serviço pagos ou no caso onde os sistemas finais receptores têm graus de importância diferentes representados através de pesos.

se $\overline{perdas}$ é $CONGESTIONADA$ então $Bps \leftarrow max(Bps_R \times \nu, Bps_{min})$
se $\overline{perdas}$ é $CARREGADA$ então $Bps \leftarrow Bps$
se $\overline{perdas}$ é $DESCARREGADA$ então $Bps \leftarrow min(Bps_R + \eta, Bps_{max})$

Figura 5.6: Base de regras para o CN.

A interface de codificação é usada pelo CN para converter $\overline{perdas}$ em graus de pertinência através de uma pesquisa na base de dados que contém as funções da pertinência associadas aos três conjuntos nebulosos. O domínio (universo de discurso) de $\overline{perdas}$ é o conjunto $\%perdas = [0, 1]$. Esse domínio contém três conjuntos nebulosos: $DESCARREGADA$, $CARREGADA$ e $CONGESTIONADA$ que representam os possíveis estados da rede. A cada conjunto nebuloso, há uma função de pertinência associada que fornece o grau de pertinência dos elementos de $\%perdas$. Por exemplo, poder-se-ia estabelecer que para um valor de perdas $x = 0.7$, $\mu_{DESCARREGADA}(x) = 0.00$, $\mu_{CARREGADA}(x) = 0.53$, $\mu_{CONGESTIONADA}(x) = 0.84$.

A saída final (já na forma não-nebulosa) é o resultado da contribuição ponderada de cada regra, sendo dada pela equação:

$$Bps = \frac{\alpha_0 \times (Bps_R + \eta) + \alpha_1 \times Bps + alpha_2 \times (\nu \times Bps_R)}{\alpha_0 + \alpha_1 + alpha_2} \tag{5.9}$$

$$\begin{aligned} \alpha_0 &= \mu_{DESCARREGADA}(\overline{perdas}) \\ \alpha_1 &= \mu_{CARREGADA}(\overline{perdas}) \\ \alpha_2 &= \mu_{CONGESTIONADA}(\overline{perdas}) \end{aligned}$$

5.4.2.3 Atuador

Além do cálculo da taxa de bits adaptada ao estado da rede, o mecanismo de adaptação necessita mapear essa taxa para a aplicação multimídia a ser adaptada a fim de que a mesma passe a enviar o fluxo na taxa calculada, isto é, é necessário definir como a nova taxa de bits será alcançada na camada de aplicação em termos de seus parâmetros de QoS. Uma vez que diferentes níveis de QoS podem ter exigências similares de taxa de bits representando, contudo, qualidades completamente distintas, o atuador deve pesquisar na tabela $\Omega_{NiveisQoS}$ o primeiro nível de QoS $L_j =< Bps_j, \rho_{1_j}, \rho_{2_j}, ..., \rho_{m_j}, Q\hat{o}S_j >$ tal que $Bps_j \leq Bps$ (uma vez que a chave de ordenação da tabela é $Q\hat{o}S$, L_j será o nível de QoS com o maior grau de qualidade que satisfaz a condição).

A seguir, o atuador altera os valores dos parâmetros de QoS da aplicação para

$\rho_{1_j}, \rho_{2_j}, ..., \rho_{m_j}$ de modo que ela passe a enviar o fluxo multimídia com o nível de QoS L_j.

A mudança do nível de QoS do fluxo multimídia pode ser efetuada de duas maneiras (Figura 5.7): filtragem (Yeadon et al. 1996) e alteração dinâmica dos parâmetros do codificador (Koliver et al. 2000 a). A primeira abordagem pode ser usada tanto no caso de aplicações de dados vivos quanto aplicações de dados armazenados; a segunda abordagem tem seu uso restrito às aplicações que usam dados vivos.

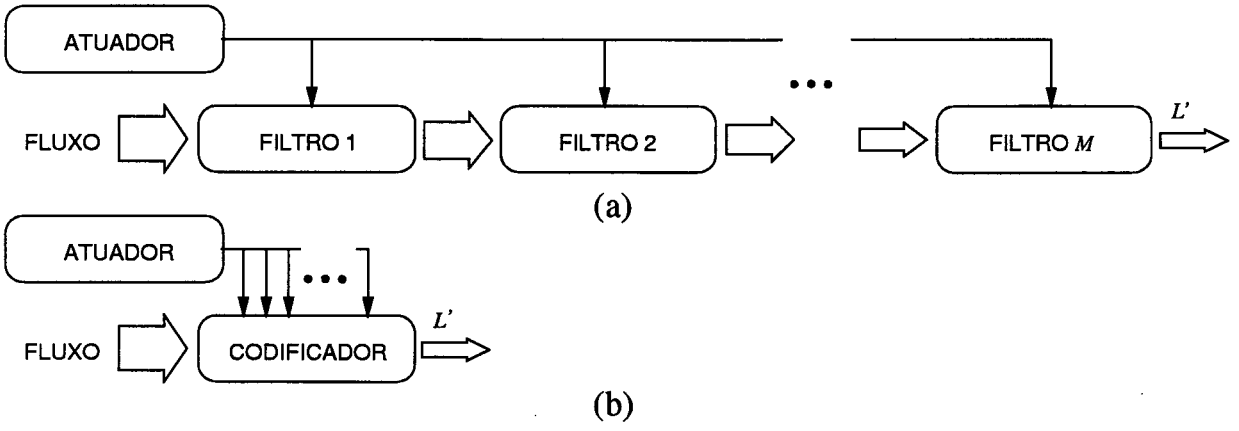

Figura 5.7: Alteração de valores de parâmetros de QoS: (a) filtragem e (b) mudança dinâmica dos parâmetros do codificador.

5.4.2.4 Comparação com Outras Propostas

O mecanismo de adaptação de QoS apresentado possui algumas similaridades com os mecanismos EACM e com o DAA, descritos no início deste capítulo. Os três usam a taxa de perdas de pacotes como variável de realimentação, fazem uso de intervalos ou faixas de valores dessa taxa quando da decisão de manter, incrementar ou decrementar a taxa de bits e usam funções lineares para calcular a nova taxa de bits. Em virtude dessas similaridades, nesta seção será feita uma comparação dos três mecanismos em relação a três aspectos: comportamento da taxa de bits, justiça ("fairness") no ajuste e escalabilidade. Visando facilitar a comparação, serão consideradas comunicações 1:1. Nesse contexto, os algoritmos de controle do EACM e DAA resumem-se aqueles mostrados nas Figura 5.8 e Figura 5.9.

Assim como no mecanismo de adaptação de QoS nebuloso, os intervalos usados pelo EACM e pelo DAA podem ser representados na forma de conjuntos, conforme a Figura 5.10. Os intervalos usados pelo EACM podem ser vistos como conjuntos nebulosos que admitem apenas

se $perdas^f \geq perdas_l$ então $Bps \leftarrow max(Bps_R \times \nu, Bps_{min})$
senão se $perdas_u \leq perdas^f < perdas_l$ então $Bps \leftarrow Bps$
senão se $perdas^f < perdas_u$ então $Bps \leftarrow min(Bps_R + \eta, Bps_{max})$

Figura 5.8: Algoritmo do EACM para apenas um sistema final receptor.

dois graus de pertinência[2], 0 ou 1. O mesmo vale para os dois intervalos usados pelo DAA. Para o CN, os conjuntos nebulosos podem ser obtidos, por exemplo, usando-se a função trapezoidal $\mu_{A_i}(x)$ associada aos parâmetros $< a_i, b_i, c_i, d_i >$ $(a_i \leq b_i \leq c_i \leq d_i)$. A função $\mu_{A_i}(x)$ é definida como:

$$\mu_{A_i}(x) = \begin{cases} 0 & \text{se } x < a_i \\ \frac{x-a_i}{b_i-a_i} & \text{se } a_i \leq x < b_i \\ 1 & \text{se } b_i \leq x < c_i \\ \frac{d_i-x}{d_i-c_i} & \text{se } c_i \leq x < d_i \\ 0 & \text{se } x \leq d_i \end{cases} \tag{5.10}$$

se $perdas^f > perdas_c$ então $Bps \leftarrow max(Bps_R \times (1 - perdas^f + perdas_c), Bps_{min})$
senão se $perdas^f \leq perdas_c$ então $Bps \leftarrow min(Bps_R + \eta, Bps_{max})$

Figura 5.9: Algoritmo do DAA para apenas um sistema final receptor.

Novamente visando facilitar a comparação das abordagens, os limites usados pelo EACM foram ajustados em $perdas_u = 0.02$ e $perdas_l = 0.04$ (valores apropriados para uma transmissão de vídeo, segundo (Busse et al. 1995)) e o limite $perdas_c$ usado pelo DAA foi ajustado em $\frac{perdas_u + perdas_l}{2} = 0.03$. No CN, os graus de pertinência para o conjunto DESCARREGADA são obtidos através da chamada $\mu_{DES}(\overline{perdas})$, sendo

$$< a_i, b_i, c_i, d_i > = < 0, 0, 0, perdas_u >;$$

para o conjunto CARREGADA, a função de pertinência usada é $\mu_{CAR}(x)$ sendo

$$< a_i, b_i, c_i, d_i > = < 0, perdas_u, perdas_l, perdas_l + perdas_u >;$$

[2]De fato, a Teoria dos Conjuntos "Clássicos" pode ser vista um subconjunto da Teoria dos Conjuntos Nebulosos.

para o conjunto CONGESTIONADA, $\mu_{CON}(x)$ sendo

$$< a_i, b_i, c_i, d_i >=< perdas_l, perdas_l + perdas_u, 1, 1 > .$$

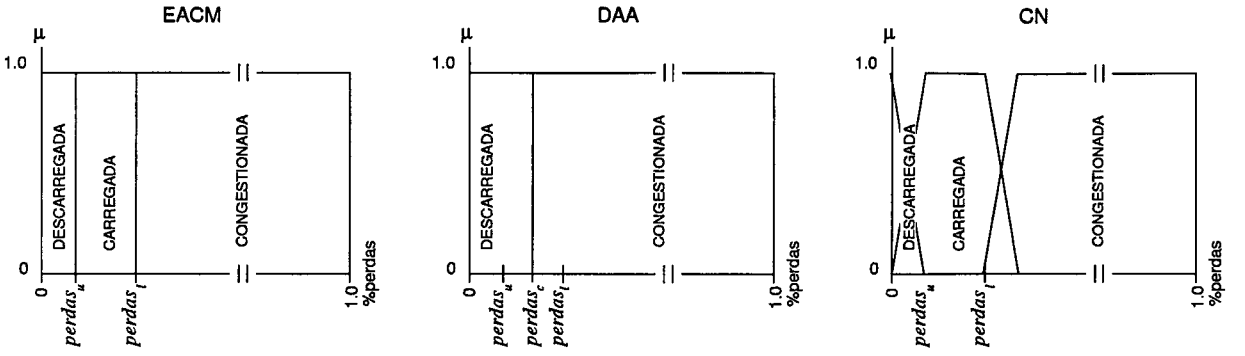

Figura 5.10: Funções de pertinência para os três mecanismos.

Essa escolha garante que o mecanismo nebuloso terá a mesma zona morta do EACM. O uso de funções trapezoidais é justificado apenas pela simplicidade; funções de pertinência mais apropriadas para determinadas aplicações podem ser obtidas usando-se, por exemplo, o método de agrupamento c-means (Bezdek 1981) ou sistemas "neuro-fuzzy´´ (vide Apêndice B).

5.4.2.4.a Comportamento da Taxa de Bits

Um mecanismo de controle de taxa de bits deve manter um compromisso entre a tentativa de determinar uma taxa de bits compatível com o estado da rede e a realização de uma adaptação suave. O cálculo de uma taxa de bits incompatível com a disponibilidade de largura de banda pode conduzir a mais perdas ou à sub-utilização da rede. Em contrapartida, uma ação de controle que implique em uma mudança brusca na taxa de bits refletir-se-á, para o usuário final, em uma mudança brusca da qualidade da apresentação. A Figura 5.11 mostra os valores da taxa de bits calculada pelos três mecanismos em função de diferentes variações de perdas ($\Delta perdas = 0, 0.01, 0.02, ..., 0.10$). O cálculo da nova taxa de bits considera que a taxa corrente, em todos os casos, é de 1000 kbps. Os parâmetros ν e η foram configurados em 0.875 e 50 kbps (os valores usados em (Busse et al. 1995) e (Sisalem 1998)). Na Figura 5.11, pode-se verificar que o incremento da taxa de bits realizado pelo EACM e DAA, nas regiões representando o estado de rede DESCARREGADA (de 0 à 2% e de 0 à 3% de perdas, respectivamente) é sempre de $\eta = 50$ kbps, já que ambos mecanismos não estabelecem qualquer relação entre o

aumento da taxa de bits e o valor da taxa de perdas. Já no CN o incremento é de η kbps apenas quando a taxa de perdas é igual a 0, diminuindo de forma linear à medida que as perdas aumentam. No DAA, a inexistência de uma zona morta faz com que o controlador sempre oscile entre a diminuição e o aumento da taxa de bits (exceto quando ele alcançar Bps_{min} ou Bps_{max}). Nos EACM e no CN, a zona morta é uma garantia de que nos períodos em que a taxa de perdas oscile entre $perdas_u$ e $perdas_l$ a taxa de bits permanecerá mais ou menos constante, garantindo também uma qualidade constante no sistema final receptor.

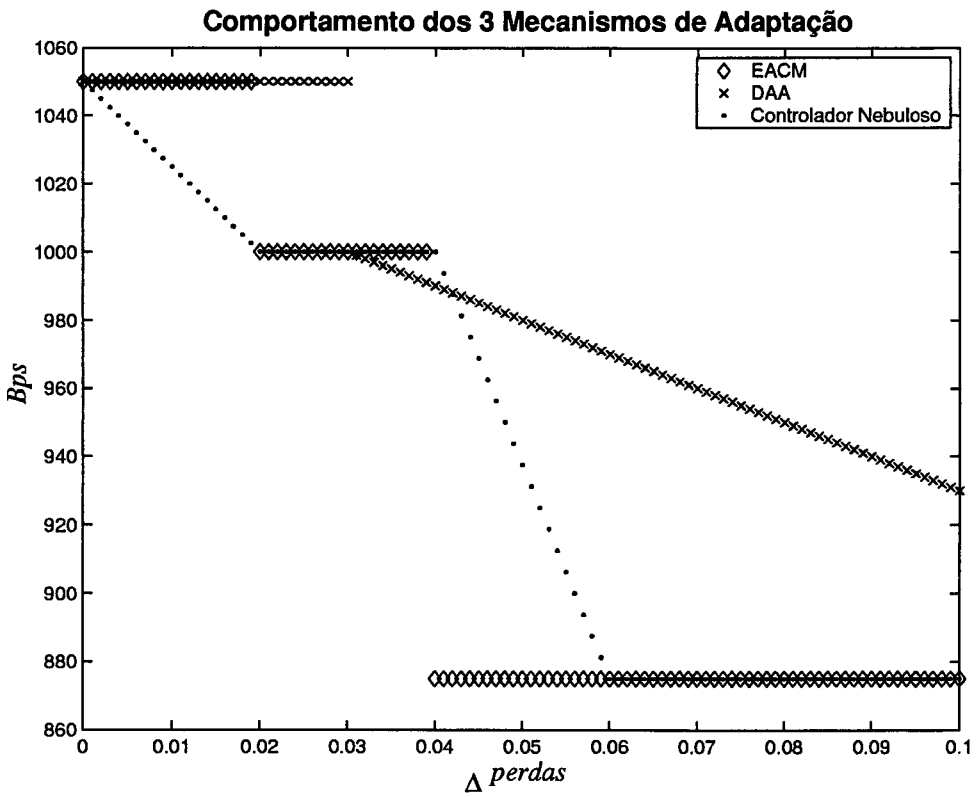

Figura 5.11: Taxa de bits calculada × variação de perdas ($Bps_r = 1000$).

A forma como os mecanismos reduzem a taxa de bits ao entrar no estado de rede CONGESTIONADA também muda bastante de uma abordagem para outra. O EACM poderá realizar uma degradação demasiadamente elevada da taxa de bits, já que ele sempre reduz a taxa à $(\nu \times 100)\%$ do valor anterior independentemente do valor da taxa de perdas. Além disso, um fator multiplicativo ν alto, aliado à uma função linear para decremento da taxa de bits que desconsidera a taxa de perdas, pode conduzir a uma sub-utilização da rede. Em contrapartida, um fator multiplicativo baixo pode fazer com que o controlador demore para encontrar uma taxa de bits compatível com a carga da rede. O DAA realiza uma degradação mais suave, de acordo

com o valor da taxa de perdas. Neste caso o problema é que, conforme o comportamento da taxa de perdas, o mecanismo pode demorar muito tempo para encontrar uma taxa de bits compatível com o estado da rede. O CN, por fim, também realiza uma degradação suave na taxa de bits, porém mais acentuada do que aquela proporcionada pelo DAA.

A suavidade no comportamento do CN (tanto no incremento quanto no decremento da taxa de bits) é decorrência do fato de que todas as regras são consideradas no cálculo da saída, com mais ou o menos peso, de acordo com o valor da entrada. *Tal característica faz com que, implicitamente, a taxa de perdas sempre seja considerada no cálculo da taxa de bits*, diferentemente do que ocorre no EACM e no DAA. Por considerar a taxa de perdas juntamente com a taxa de bits corrente no cálculo da nova taxa de bits, a decisão de ajuste é realizada de forma mais compatível com o estado da rede do que no EACM e DAA. A maneira como o CN age (com mais ou menos suavidade) pode ser facilmente ajustada mudando-se as funções de pertinência e/ou seus parâmetros. O CN pode também ser alterado para contemplar mais estados de rede através do acréscimo de mais conjuntos nebulosos (por exemplo, LEVEMENTE_CONGESTIONADA, MUITO_CONGESTIONADA etc.) ou de mais regras com modificadores lingüísticos ("pouco", "mais ou menos", "muito", "extremamente" etc.). Isso permite o acréscimo de mais regras (e funções lineares), permitindo que o CN contemple uma visão mais refinada dos estados de rede. Para cada estado, podem ser estabelecidos diferentes fatores de incremento e decremento. Esse refinamento tem como objetivo proporcionar um mecanismo de adaptação de QoS que atue de maneira mais próxima do comportamento geral do SMD.

5.4.2.4.b Justiça na Adaptação

Uma vez que a decisão do ajuste está fortemente relacionada às taxas de perdas relatadas, o mecanismo de controle deve usar uma política de agregação da variável de realimentação que tente ser o mais justa possível.

O EACM utiliza uma política de agregação baseada na maioria dos casos, decidindo aumentar, manter ou diminuir a taxa de bits de acordo com o estado da maioria dos membros. Os pontos limites N_d e N_h definem a prioridade da decisão: se $N_d = 0.1$ e $N_h = 0.1$, por exemplo, então o algoritmo aumentará a taxa de bits somente se ao menos 80% dos sistemas finais receptores estão descarregados. Assim como as políticas baseadas na média, essa política de agregação, baseada na maioria dos casos, garante que uns poucos sistemas finais receptores, conectados através de ligações com largura de banda baixa, não forçarão o sistema final emissor

a fornecer uma qualidade baixa em detrimento dos demais sistemas finais receptores. Por outro lado, essa política apresenta dois problemas. O primeiro é como definir os valores dos pontos limites; o segundo é que o controlador possibilita que sistemas finais receptores mantenham-se congestionados por longo tempo, quando eles representarem a minoria. Essa política de agregação permite que, em um dado instante, até $N \times (1 - (N_d + N_h))$ sistemas finais receptores não tenham qualquer influência na decisão de adaptação, mesmo que congestionados.

O DAA utiliza uma política de agregação baseada no pior caso que garante que o mecanismo tentará manter todos os sistemas finais receptores descongestionados. A desvantagem dessa política é que um único sistema final receptor cuja conexão está congestionada forçará uma redução na taxa de bits, prejudicando a qualidade de todos os demais e podendo, também, conduzir a uma sub-utilização da rede. Essa política faz com que $N-1$ sistemas finais receptores não tenham qualquer influência na decisão de adaptação.

Conforme visto, ambas políticas de agregação apresentam vantagens e desvantagens. Existe, ainda, uma série de variáveis relacionadas ao ambiente no qual a aplicação está inserida que influenciam no comportamento da taxa de perdas e devem, portanto, ter algum papel na escolha da política: número de membros da sessão, possibilidade ou não de reserva de recursos, tipo de rede (WAN/LAN), localização dos membros (próximos/distantes), capacidade das conexões (homogênea/heterogênea) etc.

Neste trabalho, conforme visto na Seção 5.4.2, a política de agregação deve considerar o tipo de ambiente na qual a aplicação multimídia será executada. Em redes melhor-esforço, a política de agregação proposta deve ser baseada na média geométrica. Políticas baseadas na média são, *per si*, mais interessantes no contexto , uma vez que elas possibilitam que todos os sistemas finais receptores exerçam alguma influência na decisão de adaptação. Além disso, a média geométrica possibilita uma política de agregação menos influenciada por valores reportados de perdas muito díspares da maioria dos casos do que políticas baseadas na média aritmética ou no pior caso. No caso em que a dispersão dos valores reportados é baixa, ela resulta em um valor que, de fato, representa o caso médio. Isso torna a política de agregação baseada na média geométrica bastante adequada para ambientes melhor-esforço, já que em tais ambientes, em virtude da impossibilidade de manutenção de uma qualidade mínima para todos os sistemas finais receptores, o objetivo do mecanismo de adaptação de QoS deve ser maximizar a qualidade para a *maior parte* dos sistemas finais receptores.

Em redes que permitem reserva de largura de banda, partindo-se da premissa que tal reser-

va implica em pagamento e, conseqüentemente, na obrigação da garantia de uma qualidade mínima, uma política de agregação baseada no pior caso é mais adequada, já que nenhum usuário deverá receber uma qualidade abaixo da mínima contratada (supondo que todo o grupo multiponto contratou a mesma qualidade mínima).

5.4.2.4.c Escalabilidade

Com relação à escalabilidade, o principal problema é a largura de banda requerida pelo tráfego do controle em detrimento do tráfego de dados. Quanto mais atualizada é a variável de estado, maior a largura de banda requerida pelo tráfego de controle em virtude do posicionamento do monitor no sistema final emissor. Como os três mecanismos vistos utilizam o mesmo protocolo para controle, o ponto a ser analisado é se eles consideram o tráfego de controle quando do ajuste da taxa de transmissão. O EACM claramente não preocupa-se com isso, já que seu incremento é sempre de $\eta = 50$ kbps. No DAA, por outro lado, o incremento será de, no máximo, 50 kbps (veja Equação 5.3; quanto mais membros tem a sessão, menor será o valor do incremento). A fim de considerar a largura de banda requerida pelo tráfego do controle, pode-se modificar a terceira regra do CN (vide Figura 5.6) para considerar o número de membros da sessão, utilizando, por exemplo, o fator th_{scale} como no DAA:

$$\texttt{se } \overline{perdas} \texttt{ é } DESCARREGADA \texttt{ então } Bps \leftarrow min(Bps_R + \frac{\eta}{min(N, th_{scale})}, Bps_{max})$$

5.4.2.5 Avaliação Experimental dos Mecanismos

A fim de verificar o comportamento dos três mecanismos de ajuste da taxa de bits em um ambiente como a Internet, onde as perdas não ocorrem de maneira uniforme e sim de forma brusca, foi implementada uma aplicação que emite um fluxo contínuo de um sistema final para outro usando RTP sobre UDP. A biblioteca RTP usada foi a JRTPLIB[3], uma biblioteca orientada a objetos escrita em C++.

O monitor, posicionado no sistema final emissor, recebe os pacotes RTCP da aplicação no lado do sistema final receptor e envia ao controlador a taxa de perdas de pacotes; o atuador altera a aplicação, no lado do sistema final emissor, para que ela envie o fluxo contínuo usando o RTP de acordo com a taxa de bits calculada pelo controlador. A Figura 5.12 representa os processos envolvidos na implementação e o fluxo de dados e controle existente entre eles.

[3]http://lumumba.luc.ac.be/jori/jrtplib/jrtplib.html

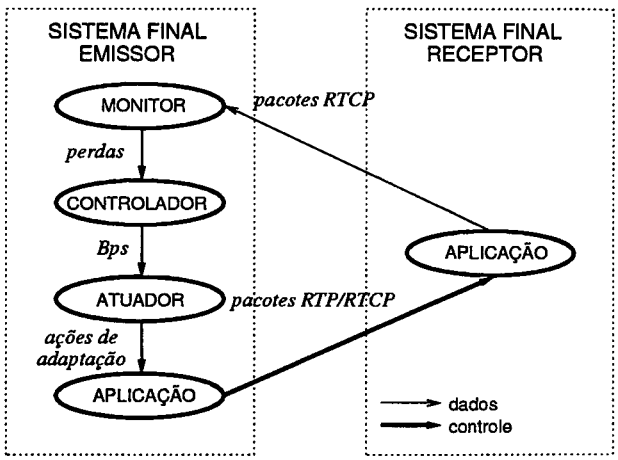

Figura 5.12: Fluxo de dados e controle.

As taxas de bits Bps_{min} e Bps_{max} foram ajustados em 50 kbps e a 5000 kbps respectivamente. O sistema final emissor usado foi uma estação de trabalho localizada na Universidade Federal de Santa Catarina, Brasil, e o sistema final receptor uma estação de trabalho na Universidade de Illinois, EUA. Os parâmetros usados nos três mecanismos foram configurados com os mesmos valores descritos na Seção 5.4.2. Para cada controlador, as medidas tiveram uma duração de 300 segundos. O parâmetro β do filtro passa baixa foi ajustado em 0.3 nos três casos (o valor mais adequado encontrado nos experimentos de Busse et al.).

De acordo com os resultados obtidos e mostrados nas Figura 5.13 e Figura 5.14, o incremento da taxa de bits[4] nos EACM (quando as perdas são menores que $perdas_u = 0.02$) e DAA (perdas menores ou iguais a $perdas_c = 0.03$) é constante, na forma de saltos de ν kbps. Já no CN o incremento é mais suave, proporcional à taxa de perdas, alcançando ν kbps apenas quando o valor das perdas é 0. Isso pode ser visto na Figura 5.15, onde os pontos representando a taxa de transmissão calculada não são igualmente espaçados. Quando da degradação, as mudanças da taxa de bits do EACM ocorrem na forma de pequenos saltos, a despeito da taxa de perdas mudar radicalmente. Isso ocorre porque quando as perdas são maiores ou iguais a $perdas_l = 0.04$, a taxa de bits passa a ser reduzida sempre em $\nu \times 100\%$. À medida que a taxa de bits é reduzida, os saltos tornam-se menores. No DAA, por outro lado, a degradação ocorre na forma de saltos maiores, proporcionais ao valor da taxa de perdas, fazendo com que a taxa de bits alcance um valor compatível com o estado da rede mais rapidamente, porém às custas de mudanças bruscas de qualidade. O CN, finalmente, realiza uma degradação de $\nu \times 100\%$ (como

[4]A taxa de bits mostrada é aquela calculada pelo controlador e não a taxa de bits real.

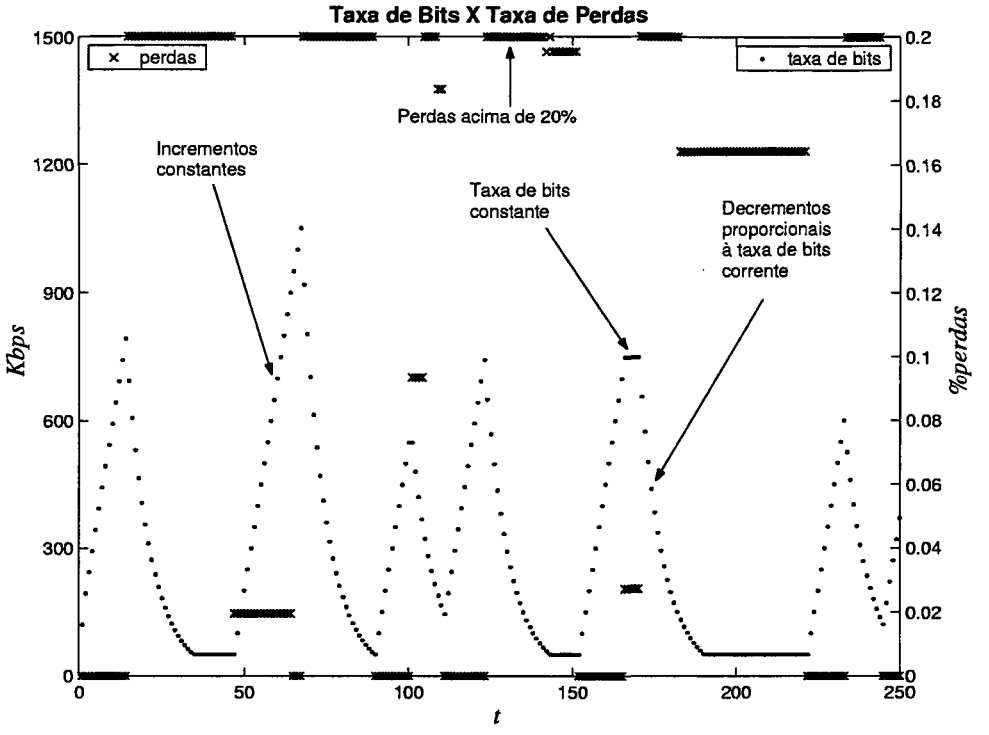

Figura 5.13: Experimentos sobre a Internet para o EACM.

aquela proporcionada pelo EACM) enquanto as perdas superam $perdas_l + perdas_u = 0.06$; para valores entre $perdas_l$ e $perdas_l + perdas_u$, a degradação é proporcional às perdas. Tal comportamento garante uma suavidade na degradação para qualquer valor de perdas.

A avaliação do desempenho dos mecanismos de controle de QoS através de grandezas mensuráveis é uma tarefa complexa. Geralmente, o parâmetro usado como critério de avaliação é a taxa de perdas de pacotes. Contudo, a despeito da taxa de perdas ser a variável de realimentação dos mecanismos vistos, ela não pode ser considerada o melhor critério, principalmente pelo fato de ser um parâmetro de QoS "distante" do ponto de vista do usuário final em virtude dos seguintes aspectos:

1. perdas baixas não indicam uma qualidade recebida alta, pois um percentual de perdas baixas pode ser obtido, por exemplo, através do envio de uma taxa de bits baixa, o que normalmente implica na transmissão de uma qualidade baixa;

2. a melhor forma de distribuição das perdas (uniformemente distribuídas ou perdas baixas com "picos"), do ponto de vista do usuário final, ainda é desconhecida, além de depender também do tipo de aplicação;

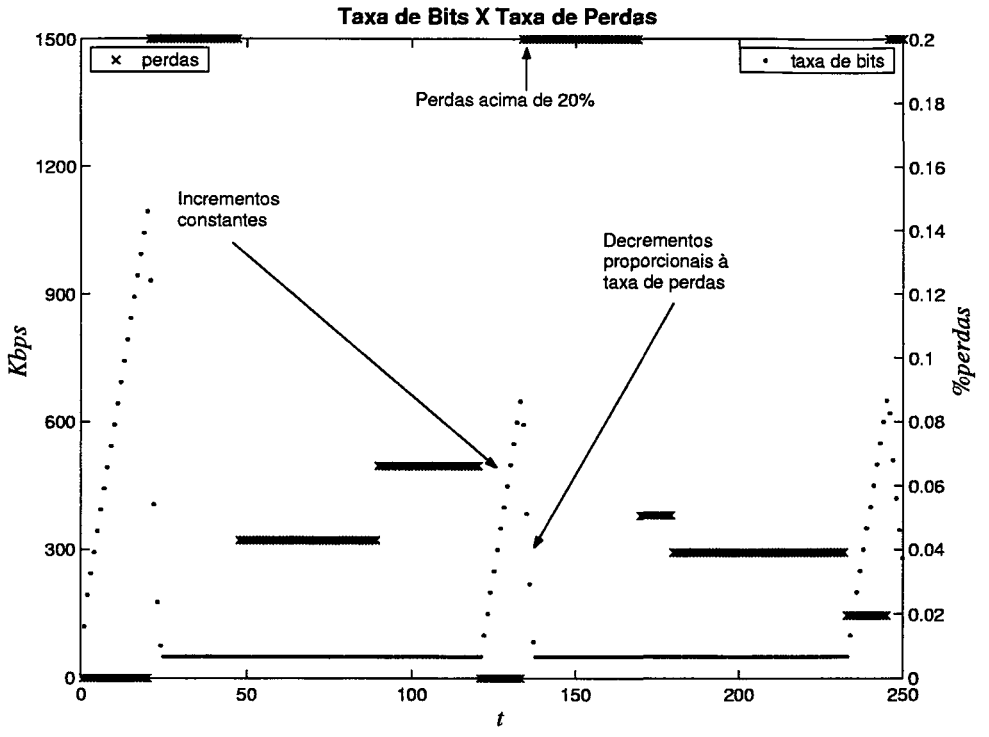

Figura 5.14: Experimentos sobre a Internet para o DAA.

3. a relação do percentual de perdas com parâmetros de QoS da camada de aplicação não é 1:1 devido ao uso dos algoritmos de compressão: a perda de um pacote contendo um quadro I, por exemplo, pode refletir-se, na camada de aplicação, na perda de vários quadros; e

4. avaliações realizadas sobre ambientes nos quais é impossível o controle da carga da rede possibilitam que as perdas mantenham-se altas mesmo após a atuação do mecanismo de controle, devido às outras aplicações executadas na rede.

Devido às limitações da taxa de perdas de pacotes (e de outros parâmetros de QoS distantes do usuário final), neste trabalho é proposto como critério de avaliação de mecanismos de adaptação de QoS o grau de qualidade recebido, obtido através do uso da função $\mathcal{QoS}$ tendo como argumentos os valores dos parâmetros de QoS recebidos. Tal proposta decorre do fato dessa métrica representar melhor o conceito de qualidade e, por conseguinte, o ponto de vista do usuário final.

Na seção seguinte, será apresentado um mecanismo de adaptação de QoS que também faz uso de um CN mas que é totalmente baseado no uso do grau de qualidade.

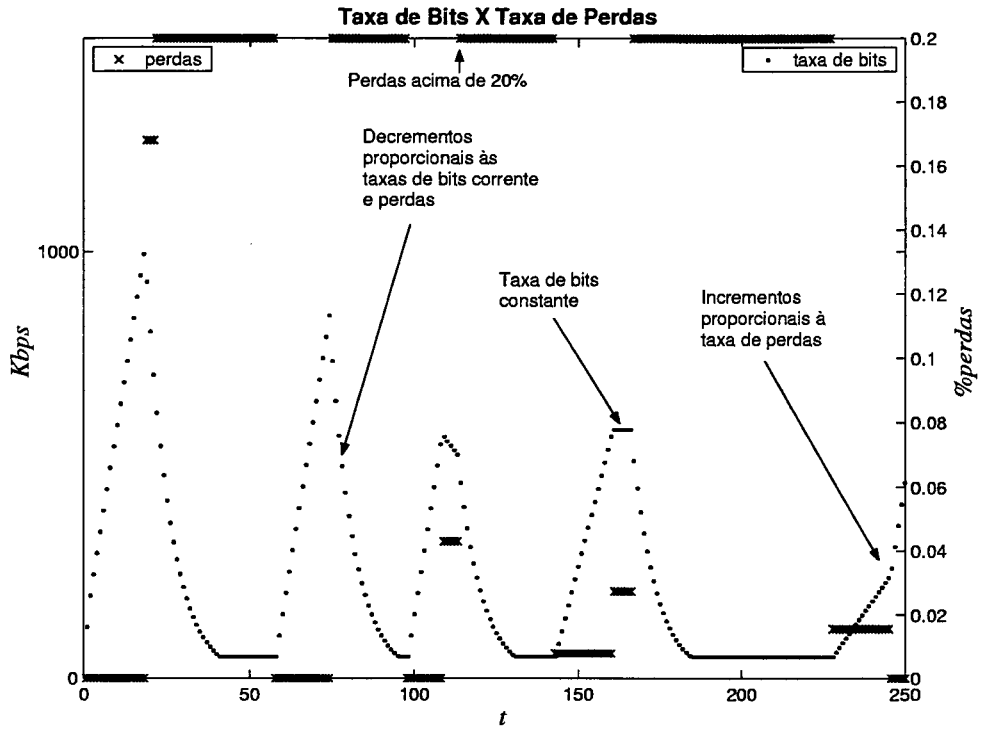

Figura 5.15: Experimentos sobre a Internet para o CN.

5.4.3 Mecanismo de Adaptação com Controle do Grau de Qualidade

Nesta seção, é descrita o segunda forma de implementação do mecanismo da adaptação de QoS proposta neste trabalho. Neste mecanismo, as variáveis de realimentação representam graus de qualidade. No modelo de adaptação de QoS, a aplicação envia um fluxo com um *grau de qualidade de emissão* $Q\hat{o}S_e$ cujo valor inicial é $Q\hat{o}S_{max} = 1.0$. O fluxo chega no $i^{\text{ésimo}}$ sistema final receptor ($i = 1, 2, ..., N$) com um *grau de qualidade de recepção* $Q\hat{o}S_{r_i}$ tal que $Q\hat{o}S_{r_i} \leq Q\hat{o}S_e$. Nos sistemas finais receptores, por causa das perdas introduzidas em decorrência da carga do processador, o fluxo é exibido com um *grau de qualidade de visualização* $Q\hat{o}S_{v_i}$ tal que $Q\hat{o}S_{v_i} \leq Q\hat{o}S_{r_i}$. Todos sistemas finais receptores realimentam o mecanismo de adaptação com $Q\hat{o}S_{v_i}$. A Figura 5.16 fornece uma visão geral do modelo de adaptação de QoS.

A Figura 5.17 é uma representação do esquema de controle. Neste mecanismo, são tratadas duas perturbações externas ao SMD: a carga da rede e a carga dos processadores dos sistemas finais. A variável de realimentação é o grau de qualidade de visualização. A variável controlada pelo CN é o grau de qualidade de emissão, calculado a partir do grau de qualidade de visualização agregado, que é mapeado pelo atuador para um nível de QoS. Como no mecanismo anterior, o atuador realiza as ações de controle sobre a aplicação multimídia para que o fluxo

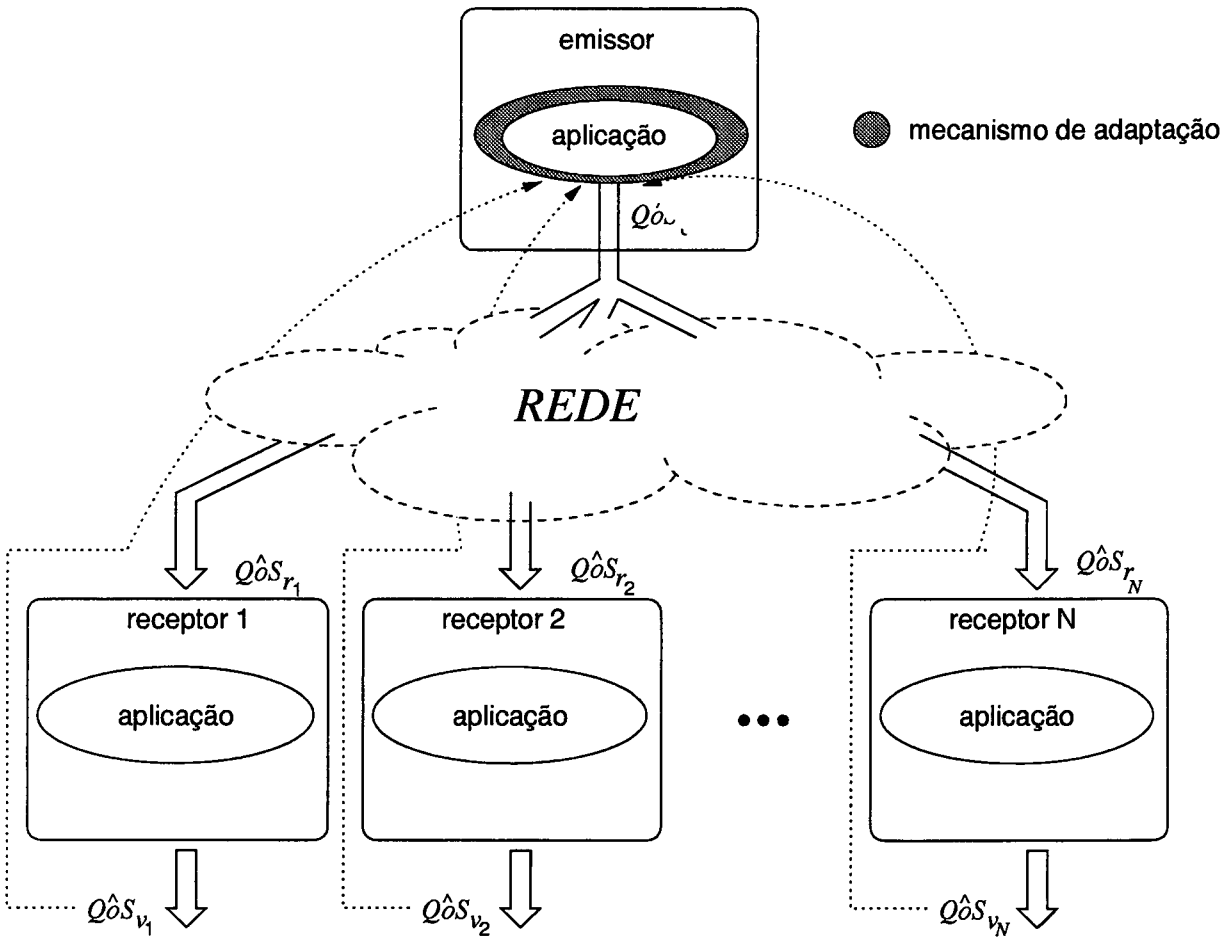

Figura 5.16: Modelo de adaptação de QoS.

passe a ser emitido com esse nível de QoS obtido. O CN e o atuador localizam-se no sistema final emissor; o monitor é distribuído nos sistemas finais receptores e no sistema final emissor. A seguir, serão detalhados os módulos do esquema de controle.

5.4.3.1 Monitor

Em cada sistema final receptor, há um monitor local que observa o grau de qualidade de visualização $Q\hat{o}S_{v_i}$ naquele sistema. Como os sistemas finais receptores enviam para o sistema final emissor o grau de qualidade de visualização e não o de recepção, a decisão de adaptação implicitamente irá considerar também a carga dos processadores desses sistemas finais, aumentando o escopo de perturbações do SMD tratadas pelo mecanismo de adaptação de QoS.

Uma vez que o nível de QoS de emissão e o nível de QoS de recepção geralmente diferem

apenas em relação à freqüência de quadros e/ou freqüência de amostras de áudio[5], para calcular o grau de qualidade de visualização basta que o monitor local observe os valores desses dois parâmetros.

Os monitores enviam os graus de qualidade de visualização para um monitor central localizado no sistema final emissor. O monitor central usa os graus de qualidade de visualização para calcular o grau de qualidade de visualização agregado $\overline{Q\hat{o}S}_v$.

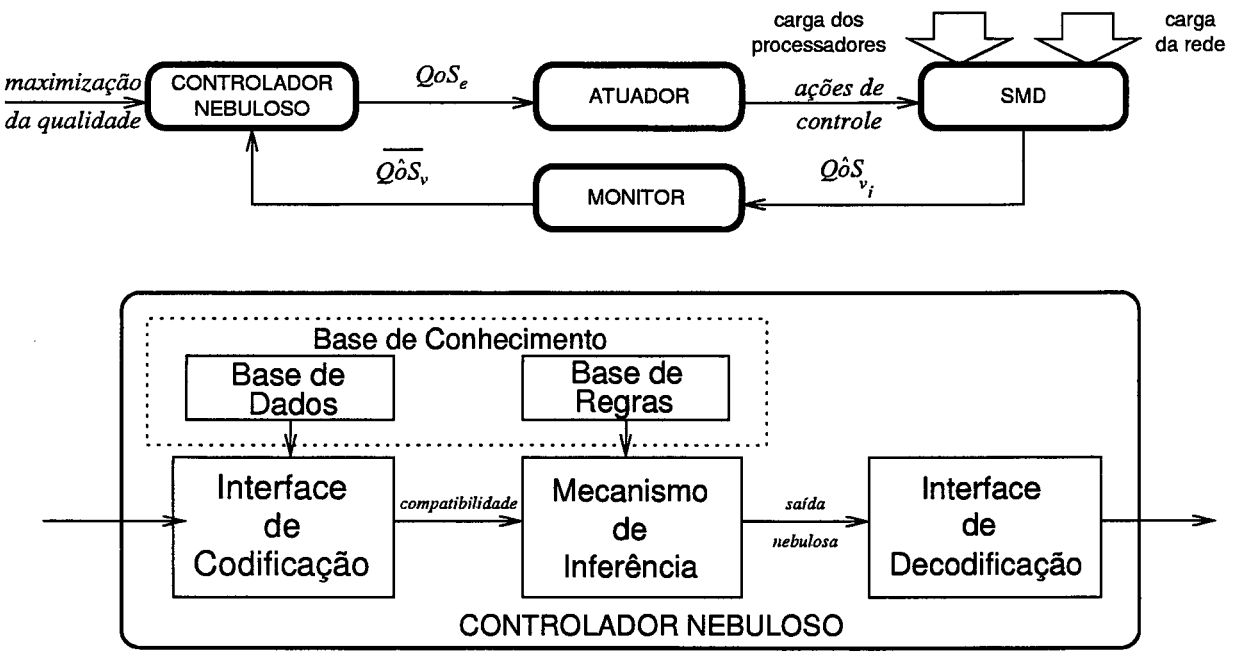

Figura 5.17: Mecanismo de adaptação.

5.4.3.2 Controlador Nebuloso

Nesta segunda forma de implementação do mecanismo da adaptação de QoS, o CN usado pelo m segue o modelo clássico de CN's. Ele recebe como entradas o grau de qualidade de emissão corrente $Q\hat{o}S_e$ e o grau de qualidade de visualização agregado $\overline{Q\hat{o}S}_v$. De posse dessas duas variáveis, o CN calcula o erro dado por:

$$\varepsilon = Q\hat{o}S_e - \overline{Q\hat{o}S}_v. \tag{5.11}$$

[5]A perda de um pacote ou do "deadline" de uma tarefa relacionada ao processamento de um dado de mídia contínua, em virtude das perturbações do SMD, implica sempre na perda de um ou mais quadros de vídeo ou amostras de áudio; apenas erros em bits - problema irrelevante em redes cujo meio físico é fibra ótica - ocasiona uma distorção na imagem. Contudo, a freqüência de quadros de vídeo e amostras de áudio é insuficiente para detectar outros aspectos relacionados à qualidade, como o tempo de resposta e presença de eco.

O sistema de inferência tem como entradas o grau de qualidade de emissão $Q\hat{o}S_e$ e a variação do erro $\Delta\varepsilon$, obtida através da seguinte equação:

$$\Delta\varepsilon = (1-\beta) \times \varepsilon(t_{n-1}) + \beta \times \varepsilon(t_n),\ 0 \leq \beta \leq 1 \tag{5.12}$$

onde $\varepsilon(t_{n-1})$ é o valor do erro obtido no instante t_{n-1} e $\varepsilon(t_{n-1})$ é o erro no instante t_n ($t_n > t_{n-1}$).

A saída do CN é decodificada gerando o novo grau de qualidade de emissão $Q\hat{o}S_e$ (Figura 5.17). Visando manter uma associação entre os conjuntos nebulosos e as cinco escalas subjetivas de qualidade para aplicações multimídia definidas pela ITU (Int 1996), o domínio do grau de qualidade ($[0, 1]$) é dividido em cinco conjuntos nebulosos: EXCELENTE, BOM, MÉDIO, POBRE e RUIM; o domínio de $\Delta\varepsilon$ é dividido em três conjuntos nebulosos: ALTO, MÉDIO e BAIXO. As regras usadas pelo sistema de inferência do CN são mostradas na Figura 5.18.

```
se QôS_e é EXCELENTE e Δε é BAIXO
então QôS_e é EXCELENTE /* Regra 0:  mantém QôS_e */
se QôS_e é EXCELENTE e Δε é MÉDIO
então QôS_e é BOM /* Regra 1:  decrementa QôS_e */
se QôS_e é EXCELENTE e Δε é ALTO
então QôS_e é MÉDIO /* Regra 2:  decrementa QôS_e */
se QôS_e é BOM e Δε é BAIXO
então QôS_e é EXCELENTE /* Regra 3:  incrementa QôS_e */
se QôS_e é MÉDIO e Δε é BAIXO
então QôS_e é BOM /* Regra 4:  incrementa QôS_e */
se QôS_e é BOM ou MEDIO e Δε é MÉDIO ou ALTO
então QôS_e é MÉDIO /* Regra 5:  mantém ou decrementa QôS_e */
```

Figura 5.18: Base de regras do CN.

5.4.3.3 Atuador

Após o CN calcular o novo grau de qualidade de emissão $Q\hat{o}S_e$, o atuador do mecanismo de adaptação procura em $\Omega_{NiveisQoS}$ todos os níveis de QoS cujos graus de qualidade sejam iguais ou maiores do que $Q\hat{o}S_e$, *i.e.*, o atuador obtém o subconjunto

$$\begin{aligned}\Omega'_{NiveisQoS} = \{ &< Bps, \rho_1, \rho_2, ..., \rho_n, Q\hat{o}S >, \\ &| Q\hat{o}S \geq Q\hat{o}S_e\}\end{aligned} \tag{5.13}$$

Dentro de $\Omega'_{NiveisQoS}$, o atuador seleciona o nível de QoS $L' =< Bps_{L'}, \rho_{1_{L'}}, \rho_{2_{L'}}, ..., \rho_{m_{L'}}, Q\hat{o}S_{L'} >$ que necessita menos largura de banda. Isso garante que a aplicação enviará o fluxo sempre com a melhor qualidade e menor consumo de largura de banda possíveis. Caso a tabela $\Omega_{NiveisQoS}$ já tenha sido otimizada de acordo com as regras descritas na Seção 5.4.1, o novo nível de QoS de emissão será o primeiro cujo grau de qualidade $Q\hat{o}S_{L'}$ é menor ou igual a $Q\hat{o}S_e$.

O processo de adaptação culmina com o atuador alterando, através de mudança nos parâmetros do codificador ou filtragem, os valores dos parâmetros de QoS da aplicação para $\rho_{1_{L'}}, \rho_{2_{L'}}, ..., \rho_{n_{L'}}$ de modo que ela passe a enviar o fluxo multimídia com o nível de QoS L'.

5.4.3.4 Experimentos

O mecanismo proposto da adaptação de QoS foi implementado a fim de controlar a qualidade de uma aplicação de distribuição de vídeo que foi desenvolvido por Yeadon et al. (Yeadon et al. 1996) visando testar os efeitos quantitativos de diferentes operações de filtragem. A aplicação original consiste de um número de componentes de "software" responsáveis pela transmissão, filtragem e exibição do fluxo de vídeo, através do reprodutor de vídeo ("video player") MPEG-1 desenvolvido pela Universidade de Berkeley (Rowe e Smith 1992). Toda comunicação é baseada em soquetes e para cada sistema final receptor, é gerada uma réplica do fluxo multimídia.

A diferença da implementação em relação ao mecanismo de adaptação de QoS baseado no grau de qualidade proposto nesta seção é que, devido às características da aplicação original, cada réplica i do fluxo de vídeo exige um CN particular. O monitor do $i^{ésimo}$ sistema final receptor envia para seu monitor global associado (localizado no sistema final emissor) a variável de realimentação. Essa variável é a freqüência de quadros exibida no sistema final receptor ou Fps_{v_i}. A freqüência de quadros é suficiente para que o $i^{ésimo}$ monitor global determine o valor de $Q\hat{o}S_{v_i}$ porque ela é o único parâmetro de QoS cujo valor recebido pode diferir daquele enviado. O $i^{ésimo}$ CN calcula o novo grau de qualidade de emissão $Q\hat{o}S_{e_i}$. Esse grau é fornecido para o atuador que altera os parâmetros dos filtros para que a $i^{ésima}$ réplica passe a ser emitida com esse grau de qualidade. A Figura 5.19 representa os processos envolvidos na implementação e o fluxo de dados e controle existente entre eles.

Os CN's calculam o novo grau de qualidade de emissão usando uma base de regras igual àquela mostrada na Figura 5.18 (todas regras têm o mesmo peso). Por simplicidade, foram

usadas funções de pertinência triangulares (Figura 5.20). O processo de decodificação usa o método do centro de gravidade (COG; vide Apêndice B).

Os atuadores usam quatro filtros para que o novo nível de QoS seja concretizado: o filtro de supressão de quadros, usado para descartar determinados quadros mudando assim a freqüência; um filtro passa-baixa que permite descartar os coeficientes DCT de mais alta freqüência dos quadros; o filtro de requantização, que permite mudar o fator de quantização; e o filtro de suavização. Uma descrição mais detalhada desses filtros pode ser encontrada em (Yeadon et al. 1996).

5.4.3.4.a Níveis de QoS e Função Grau de Qualidade

Os níveis de QoS são representados pela freqüência de quadros (*Fps*), coeficiente DCT (*lp*), fator de quantização (*q*) e fator de suavização (*smooth*), onde os três últimos parâmetros são usados pelo algoritmo de compressão MPEG-1 e influenciam a definição da imagem. O domínio desses parâmetros é:

$$\begin{aligned} \Omega_{Fps} &= \{0, 1, 2, ..., 29, 30\} \\ \Omega_{lp} &= \{1, 2, 3, ..., 62, 63\} \\ \Omega_{q} &= \{0, 1, 2, ..., 15, 16\} \\ \Omega_{smooth} &= \{0, 1, 2, ..., 19999, 20000\} \end{aligned}$$

Para o parâmetro q, 0 representa a maior qualidade e 16 a menor. Uma vez que as diferenças entre muitos valores desses parâmetros não são perceptíveis e visando reduzir o número de amostras a serem avaliadas, os domínios de *lp*, *q* e *smooth* foram reduzidos à

$$\begin{aligned} \Omega_{lp} &= \{9, 18, 27, 36, 45, 54, 63\} \\ \Omega_{q} &= \{0, 2, 4, 6, 8, 10, 12, 14, 16\} \\ \Omega_{smooth} &= \{1000, 2000, 3000, ..., 20000\} \end{aligned}$$

A escolha desse conjunto de parâmetros de QoS é devida à disponibilidade de filtros para alteração dinâmica dos valores dos mesmos oferecia pela aplicação original.

A função grau de qualidade foi obtida através da abordagem 1 descrita na Seção 3.3.3 do Capítulo 3. Nessa abordagem, o grau de qualidade de um nível de QoS é obtido a partir das funções utilidade obtidas para cada parâmetro individual. A Figura 5.21 mostra as funções utilidade para os quatro parâmetros de QoS considerados. As curvas foram obtidas por interpolação polinomial sobre pontos de inflexão arbitrados de forma intuitiva obtidos a partir da observação

de três clipes MPEG-1 diferentes. Para o nível de QoS $< \dot{Fps}, \dot{lp}, \dot{q}, \dot{smooth} >$, o grau de qualidade será igual a $min(\upsilon_{Fps}(\dot{Fps}), \upsilon_{lp}(\dot{lp}), \upsilon_{q}(\dot{q}), \upsilon_{smooth}(\dot{smooth}))$, onde $\upsilon_{\rho_i}(\dot{\rho}_i)$ é a utilidade do parâmetro ρ_i.

5.4.4.4.b Desempenho

O mecanismo implementado foi executado sobre uma rede local com quatro máquinas. O sistema final emissor usado foi uma máquina Sun Sparc (Boston; SUNsparc Ultra-60; com uma RAM de 1024 mbytes), os sistemas finais receptores também eram três máquinas Sun Sparc: Lima (SUNsparc Ultra-2; RAM de 128 Mbytes), Athena (SUNsparc Ultra-2; RAM de 128 mbytes) e Paris (SUNsparc Ultra-60; RAM de 1024 Mbytes). O gráfico de Figura 5.22 compara o desempenho da aplicação para uma comunicação 1:1 realizada entre as máquinas Boston e Athena, sem e com controle de QoS.

O momento em que a rede é sobrecarregada (usando o programa Netperf[6]) é o momento em que a taxa de bits, perdas, freqüência de quadros e o grau de qualidade mudam drasticamente de valores. Como pode ser visto, a aplicação apresentou um desempenho muito melhor tanto considerando-se dois critérios comuns da avaliação (a taxa das perdas de quadros e a freqüência de quadros) quanto em termos do critério de avaliação proposto neste trabalho (grau da qualidade). Entretanto, com controle de QoS, o comportamento é menos estável, provavelmente porque a aplicação demora a responder às ações dos filtros.

A Figura 5.22 mostra os resultados obtidos para taxa de bits, taxa de perdas de quadros, freqüência de quadros (exibida) e grau de qualidade diante de perturbações de carga da rede sem e com adaptação de QoS. Os gráficos referentes aos graus de qualidade e a taxa de bits dessa figura corroboram a afirmação de que uma maior taxa de bits não fornece, necessariamente, uma maior qualidade: a diferença entre as taxas de bits da aplicação com e sem controle não é significativa (em média, 1500 kbps antes do aumento da carga e 500 kbps após ao aumento); entretanto, o grau de qualidade difere bastante entre uma situação e outra: 0.6 (antes do aumento da carga) 0.05 (após o aumento da carga) sem controle de QoS e 0.9 (antes do aumento da carga) e 0.5 (após do aumento da carga) com controle de QoS. Isso respalda a necessidade da existência de alguma forma de mapeamento entre taxa de bits, parâmetros de QoS e qualidade, o que é feito neste trabalho através da função grau de qualidade. Por fornecer a melhor combinação de valores de parâmetros de QoS para uma dada largura de banda, a função grau de qualidade

[6]http://www.netperf.org

permite que essa largura de banda seja melhor aproveitada em termos de qualidade. Em virtude disso, mecanismos de adaptação de QoS que atuam sobre um único parâmetro de QoS podem não ser satisfatórios sob o ponto de vista do usuário final, dependendo da disponibilidade de largura de banda. Por exemplo, um mecanismo cuja atuação ocorra somente sobre o fator de quantização pode levar este parâmetro de QoS a valores que representem uma qualidade muito baixa (especialmente em uma rede melhor-esforço). Tal situação pode ser evitada se a degradação é balanceada entre outros parâmetros, como a freqüência de quadros, a profundidade do pixel e a freqüência de amostras de áudio, além de outros parâmetros usados pelo codificador.

O grau de qualidade, por sua vez, é uma variável de realimentação tão adequada para identificar a carga da rede quanto outras da camada de comunicação, como mostram os gráficos referentes a taxa de bits, perdas e grau de qualidade com controle de QoS da Figura 5.22. Tanto nos períodos de congestionamento quanto nos períodos em que a rede está apenas carregada, a taxa de bits reage rapidamente às mudanças nos valores do grau de qualidade de visualização. Essa mudança na taxa de bits ocasiona uma queda na taxa de perdas de quadros. A vantagem do uso do grau de qualidade como variável de realimentação sobre outros parâmetros de QoS é que, além de permitir que o mecanismo de adaptação tenha uma idéia do estado da rede, ele realmente mostra como as perdas estão sendo sentidas pelo usuário final, o centro de todo o processo de adaptação de QoS segundo a concepção deste trabalho.

As Figuras 5.23, 5.24, 5.25 e 5.26 mostram os resultados médios alcançados com três sistemas finais receptores para a taxa de bits, taxa de perdas de quadros, freqüência de quadros e grau de qualidade. A diferença de desempenho com e sem controle de QoS diminuiu porque as operações de filtragem foram executadas concorrentemente no sistema final emissor, aumentando o tempo de resposta do controlador. Tal problema, contudo, tende a ser minimizado rapidamente, com o crescente aumento de poder de processamento dos computadores.

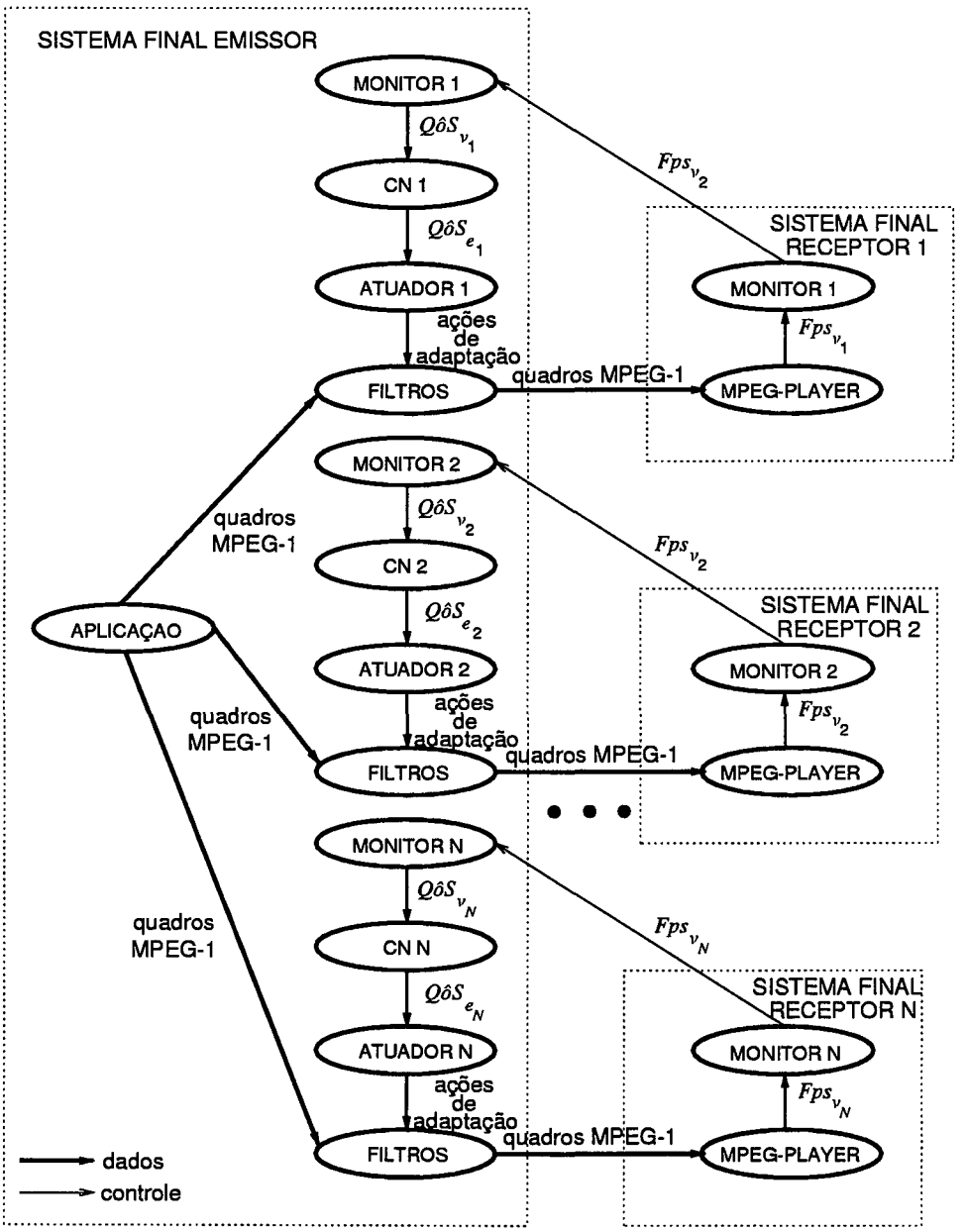

Figura 5.19: Fluxo de dados e controle.

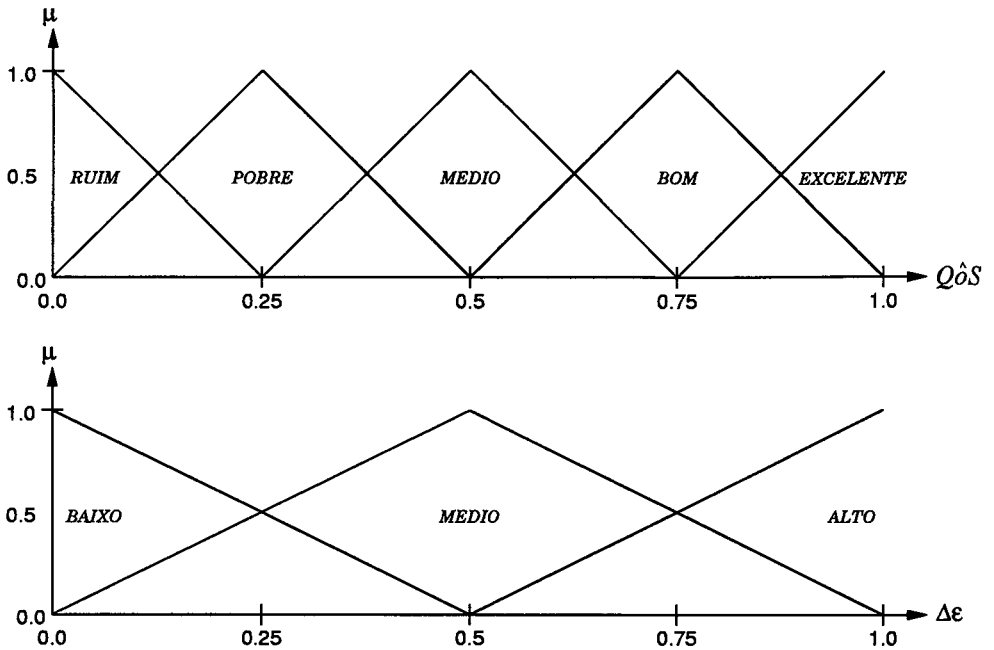

Figura 5.20: Funções de pertinência usadas.

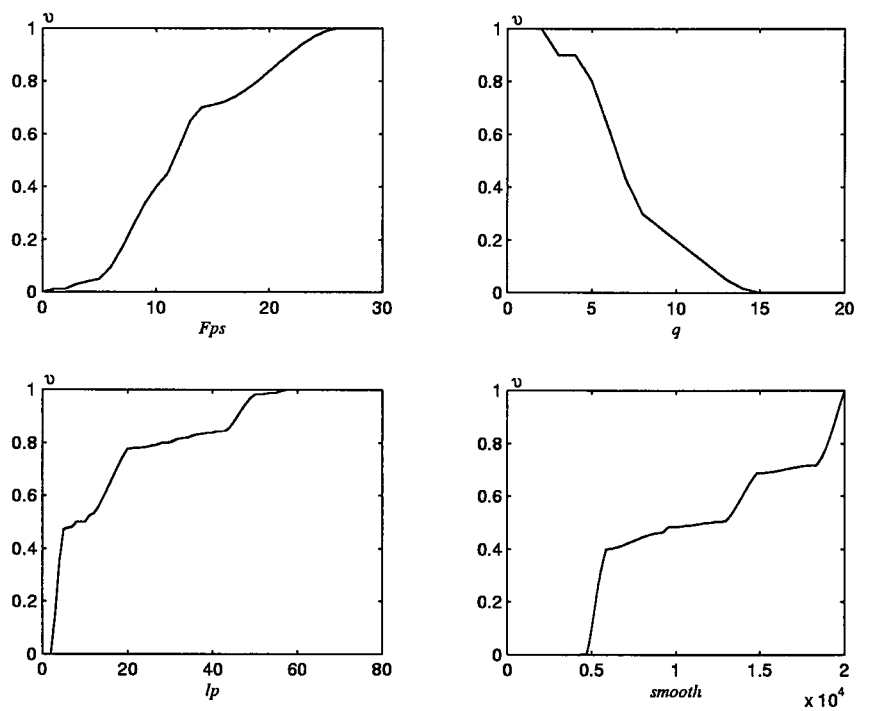

Figura 5.21: Funções utilidade para a freqüência de quadros (Fps), coeficiente DCT (lp), fator de quantização (q) e fator de suavização ($smooth$).

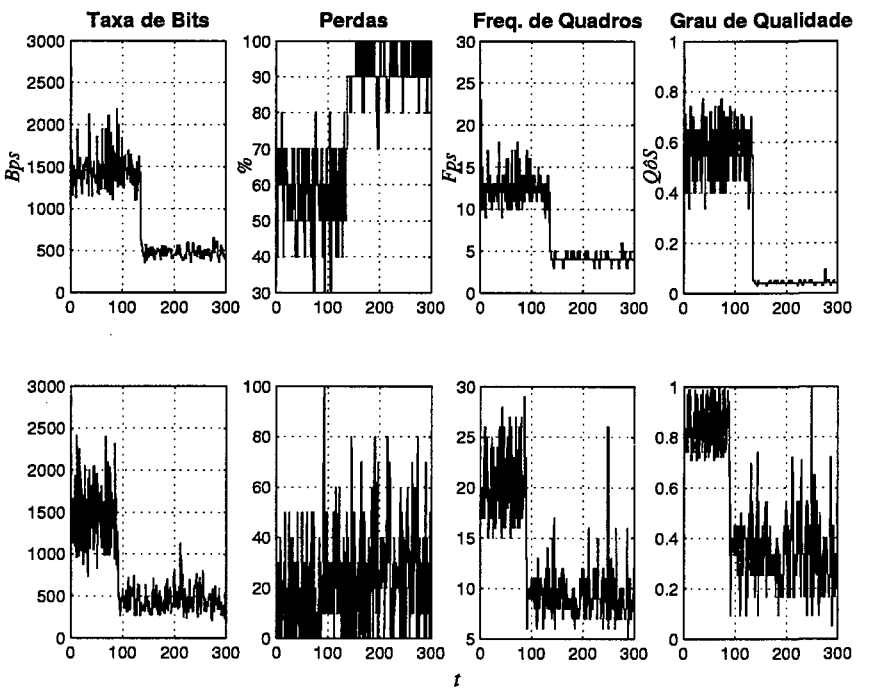

Figura 5.22: Desempenho da aplicação para comunicações 1:1: sem controle de QoS (acima) e com controle de QoS (abaixo), para os casos de rede carregada e congestionada.

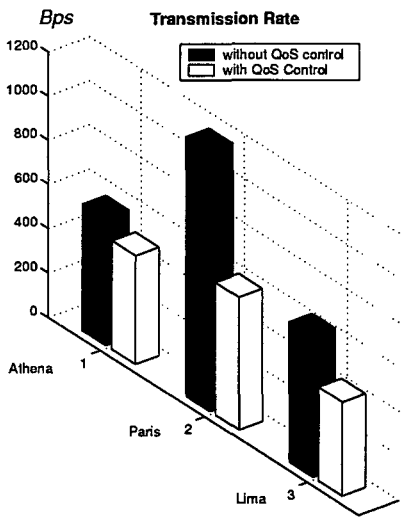

Figura 5.23: Taxa de bits.

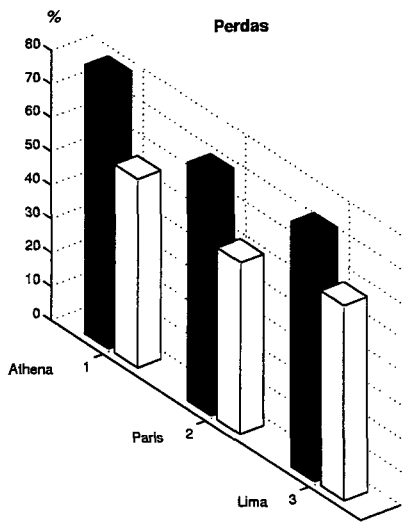

Figura 5.24: Taxa de perdas de quadros.

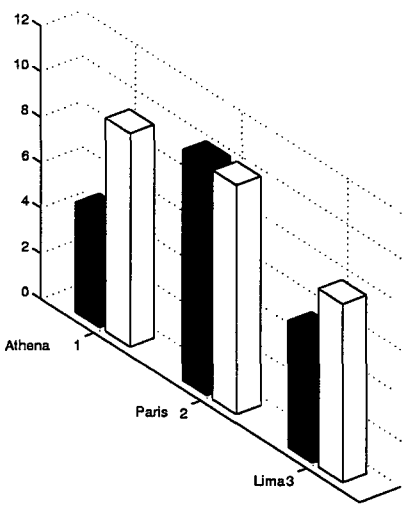

Figura 5.25: Freqüência de quadros.

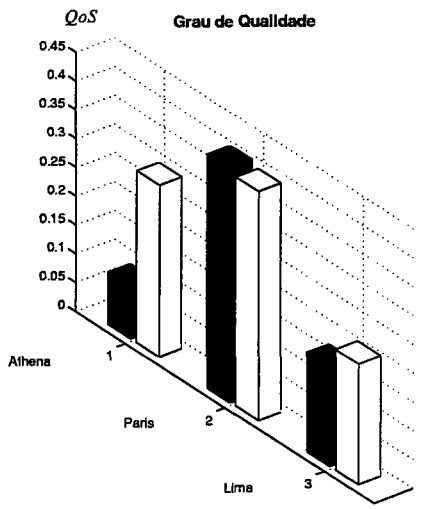

Figura 5.26: Grau de qualidade.

5.4.3.4.c Complexidade Computacional

A complexidade computacional do mecanismo de adaptação de QoS é baixa devido ao número reduzido de regras. Assim, o controle de QoS não adiciona uma carga significativa à máquina obstante o número de sistemas finais receptores. De fato, a maior carga é adicionada pelas operações de filtragem. Idealmente, essas operações deveriam ser executadas via "hardware", o que diminuiria o tempo de resposta do mecanismo de adaptação, aumentando sua eficiência. A Figura 5.27 mostra a variação da carga do processador do sistema final emissor para uma

comunicação 1:5 (o topo indica 100% de uso do processador). A carga do processador chega a 50% quando a aplicação é iniciada porque a aplicação original exige que uma instância do processo do reprodutor de vídeo MPEG-1 seja disparada no sistema final emissor. A carga tem um leve aumento quando os clientes nos sistemas finais receptores são iniciados. O controle, quando restrito à monitoração e ao cálculo de $Q\hat{o}S_{e_i}$, aumenta a carga apenas durante o processo de inicialização. Entretanto, quando iniciam as operações de filtragem, a carga do processador aumenta em torno de 70%, permanecendo alta durando todo o transcorrer da sessão da aplicação. Esse comportamento mostra a inviabilidade do uso de filtragem via "software" para muitos sistemas finais receptores. Contudo, deve-se salientar que em virtude das características da aplicação de distribuição de vídeo utilizada, muita da carga adicional é devida à necessidade de replicação do fluxo ainda no sistema final emissor, o que exige a individualização da filtragem para cada réplica.

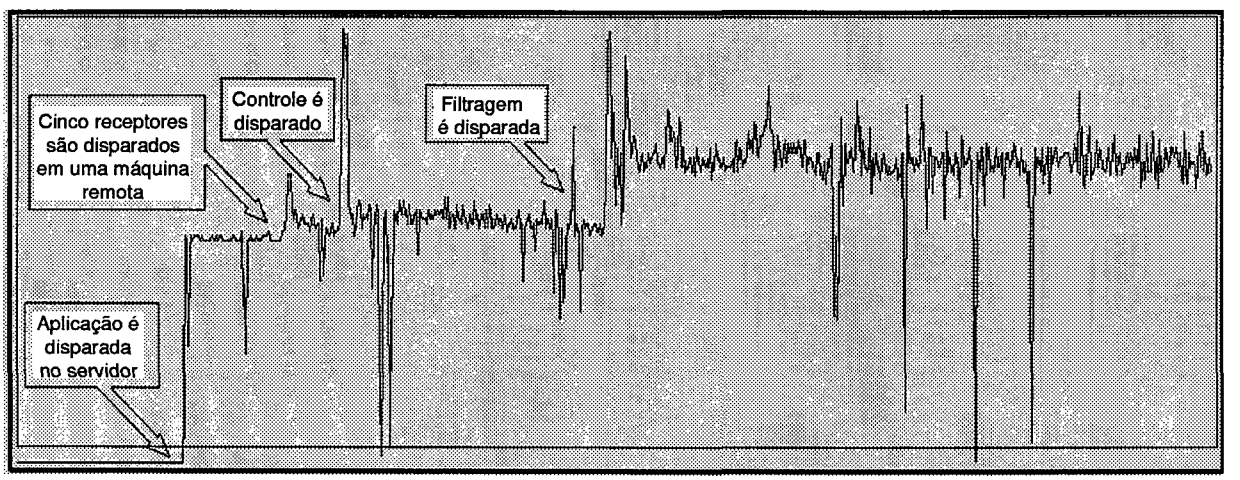

Figura 5.27: Taxa de uso do processador.

5.5 Resumo e Discussão

Neste capítulo foi apresentado um modelo de adaptação de QoS baseado no uso de controle nebuloso bem como os resultados obtidos através de implementações desse modelo. Duas versões do mecanismo foram propostas, uma baseada no controle da taxa de bits e outra baseada no controle do grau de qualidade. A primeira versão foi implementada utilizando o modelo de controle nebuloso por interpolação e utilizando um protocolo de comunicação (RTP/RTCP) já consolidado para o transporte e controle de aplicações multimídia. Por outro lado, ela não permite

que o mecanismo de adaptação de QoS tenha ciência da qualidade que os usuários estão experimentando. A segunda versão, implementada usando o modelo de controle nebuloso clássico, é mais direcionada para o objetivo do mecanismo (a maximização da qualidade percebida pelos usuários). Ela contempla, além da carga da rede, outra perturbação do SMD, a carga dos processadores e permite a construção de um controlador de forma intuitiva e que pode ser facilmente refinado. Em contrapartida, ela exige a criação de canais adicionais para a realimentação.

As implementações de ambas versões confirmaram a adequação do uso de controle nebuloso em mecanismos de adaptação de QoS. O uso do CN proporcionou uma adaptação mais suave, oferecendo, ainda, diferentes possibilidades, representadas através de diferentes sistemas de inferência, de acordo com a disponibilidade de informações que estabeleçam uma relação entre parâmetros de QoS (como taxa de perdas de pacotes, taxa de ocupação de "buffers"e atraso) e o estado da rede.

A função grau de qualidade, por sua vez, preenche a lacuna existente na maioria das propostas de mecanismos de adaptação de QoS relacionada ao mapeamento de parâmetros de QoS, qualidade (como um todo) e necessidades de recursos. Todavia, para que isso ocorra de fato, sua construção deve ser embasada em dados obtidos a partir de entrevistas realizadas sobre universos heterogêneos de usuários.

Assim como todas as abordagens de adaptação de QoS baseadas em único controlador posicionado no sistema final emissor e realimentado pelos sistemas finais receptores, o mecanismo de adaptação descrito neste trabalho pode apresentar problemas relacionados à escalabilidade e à justiça ("fairness") na adaptação. No caso da escalabilidade, o problema que pode vir a ocorrer é uma explosão de realimentação, isto é, o próprio mecanismo de adaptação pode congestionar a rede quando N (o número de sistemas finais receptores) é muito grande.

Outro problema relacionado à escalabilidade (e à distância entre os sistemas finais) diz respeito ao atraso no recebimento da variável de realimentação. Tal atraso pode comprometer a eficiência do mecanismo de adaptação de QoS por obrigá-lo a trabalhar com dados defasados.

No caso da justiça, o problema está relacionado à política de agregação usada para o cálculo da variável de decisão. Políticas de agregação baseadas em média permitem que sistemas finais receptores continuem sofrendo perdas indefinidamente; políticas baseadas no pior caso permitem que sistemas finais receptores recebam um qualidade muito aquém de suas possibilidades.

Em virtude dos problemas acima mencionados, o modelo de adaptação de QoS foi esten-

dido para contemplar o uso de vários controladores espalhados nos nós intermediários da rede (roteadores, "switches" etc.) ao invés de um único no sistema final emissor. A descrição desse modelo de adaptação de QoS com controle distribuído será feita no próximo capítulo.

Capítulo 6

MODELO DE ADAPTAÇÃO DE QoS DISTRIBUÍDO

6.1 Introdução

Conforme visto no capítulo anterior, as abordagens de adaptação de QoS baseadas em troca de mensagens de controle entre sistemas finais possibilitam a ocorrência de explosão de realimentação. Outro problema comum às abordagens (independentemente de serem baseadas em laço aberto ou laço fechado) é a questão da justiça na adaptação, já que qualquer política de adaptação possibilita que determinados sistemas finais receptores (e usuários, em uma última instância) sejam prejudicados pelas ações de adaptação no sentido de receber uma qualidade degradada, quando o sistema final emissor reduz a taxa de bits abaixo das capacidades das ligações das sub-árvores associadas a esses sistemas finais receptores ou quando ele envia uma taxa de bits acima das capacidades das ligações das sub-árvores.

Além dos dois problemas supramencionados, um terceiro problema das abordagens baseadas em trocas de mensagens entre sistemas finais é o atraso no recebimento dessas mensagens. Esse problema é particularmente crítico em WAN's, com grandes distâncias entre os nós, e em redes melhor-esforço, nas quais congestionamentos podem fazer com que mensagens de controle aguardem muito tempo nos "buffers" ou mesmo sejam perdidas. Esse atraso pode fazer com que o mecanismo de adaptação demore muito tempo para reagir às perturbações do SMD ou que ele trabalhe com variáveis de realimentação defasadas.

Neste capítulo, será descrita a proposta de um modelo de adaptação de QoS baseado na distribuição de CN´s nos nós intermediários da rede. A distribuição tem como finalidade min-

imizar a possibilidade de ocorrência de explosão de realimentação, tornar mais justa a decisão de adaptação e reduzir o tempo de reação do mecanismo de adaptação.

6.2 Trabalhos Relacionados

Há poucos trabalhos que propõem um modelo de adaptação distribuído. Fisher et al. (Fischer, Salem e Bochmann 1997) (Fischer, Hafid, Bochmann e de Meer 1997) propõem um modelo de QoS no qual *agentes de QoS*, entidades posicionadas nos roteadores e sistemas finais dos participantes da sessão, cooperam entre si visando oferecer a QoS solicitada pela aplicação. A adaptação ocorre quando sistemas finais juntam-se ou deixam o grupo de receptores da aplicação ou quando a QoS negociada não pode mais ser suportada em determinado nó $R_{i,j,k}$[1] (os autores não especificam qual é a variável monitorada pelos agentes para avaliar a QoS). Quando ocorre a segunda situação, o agente do nó $R_{i,j,k}$ envia uma mensagem para o agente do nó $R_{i,j}$ imediatamente acima dele na árvore multiponto relatando que houve uma violação de QoS. Este, por sua vez, aguarda durante algum tempo para ver se os agentes dos outros nós abaixo dele também irão relatar violação de QoS. Se isso não ocorre, ele envia uma mensagem para o agente do nó $R_{i,j,k}$ informando que ele deverá solucionar o problema localmente, selecionado uma QoS mais baixa. Contudo, se outros agentes também relatam violação de QoS, $R_{i,j}$ envia uma mensagem para o agente do nó R_i acima dele. Este último realiza, então, uma reconfiguração parcial da árvore multiponto. O trabalho acima enfoca apenas aspectos arquiteturais e a política de adaptação, sem descrever os mecanismos utilizados para implementá-la.

O mecanismo proposto em (Bogatinovski et al. 1998), brevemente descrito no capítulo anterior, calcula o valor da variável de realimentação nos nós intermediários da rede, mas a adaptação é centralizada no sistema final emissor.

6.3 Política de Agregação Distribuída

Os problemas da possibilidade de explosão de realimentação e justiça na adaptação não possuem soluções ótimas, ou seja, uma solução que garanta a não-ocorrência de explosão de realimentação e uma decisão de adaptação que contemple completamente as necessidades de

[1] Esta notação não é a utilizada por Fisher et al. Neste capítulo, o subscrito é utilizado para indicar o posicionamento do nó na árvore multiponto. O nó $R_{\beta_1,\beta_2,\ldots,\beta_{n-1},\beta_n}$ situa-se no nível n da árvore multiponto. Imediatamente acima dele está o nó $R_{\beta_1,\beta_2,\ldots,\beta_{n-1}}$; mais acima, o nó $R_{\beta_1,\beta_2,\ldots,\beta_{n-2}}$ e assim por diante.

todos os sistemas finais receptores. A possibilidade de ocorrência de explosão de realimentação pode ser diminuída através da diminuição da freqüência das mensagens de controle (e, conseqüentemente, a largura de banda necessária para a realimentação) de controle. Essa é a abordagem usada por alguns mecanismos de adaptação de QoS, como aquele descrito em (Sisalem 1998). Tal solução, contudo, é inadequada em ambientes muito dinâmicos, pois faz com que as variáveis de realimentação cheguem ao sistema final emissor muito defasadas, fazendo com que o mecanismo de adaptação execute ações que não condizem com o contexto corrente do SMD.

Para tornar o processo de adaptação mais justo, neste trabalho é proposta uma estratégia para diminuir a distância entre os nós que enviam as taxas de perdas e o nó que calcula a taxa de perdas agregada. Tal estratégia, consiste no posicionamento de mecanismos que implementem a política de agregação nos nós intermediários da rede (roteadores, "switches", "gateways" etc.). Neste caso, um nó qualquer $R_{i,j}$ com n nós $R_{i,j,k}$ ($k = 1, 2, ..., n$) imediatamente abaixo dele na árvore multiponto enviará para o nó R_i imediatamente acima dele apenas uma mensagem de controle $< n, \overline{R_{i,j}} >$ onde $\overline{R_{i,j}} = \psi(\overline{R_{i,j,1}}, \overline{R_{i,j,2}}, ..., \overline{R_{i,j,n}})$ é a variável de realimentação agregada do nó $R_{i,j}$, $\overline{R_{i,j,k}}$ é a variável de realimentação agregada do nó $R_{i,j,k}$ e ψ é uma função que define a política de agregação da variável de realimentação. A política de agregação pode ser executada nos nós intermediários da rede através de cápsulas[2].

A função ψ pode ser a média geométrica ponderada, onde o número n de sistemas finais associado a cada variável de realimentação agregada recebida funciona como o peso no cálculo.

Para ilustrar o funcionamento, sejam seis sistemas finais receptores ($N = 6$) em uma árvore multiponto como aquela mostrada na Figura 6.1 e seja o grau de qualidade de visualização $Q\hat{o}S_v$ a variável de realimentação. $Q\hat{o}S_v$ é o grau de qualidade na exibição dos dados, ou seja, aquele realmente percebido pelo usuário final. Ele é o grau de qualidade correspondente ao nível de QoS que está sendo reproduzido no sistema final, podendo ser menor do que recebido em virtude de quadros ou amostras de áudio descartados por terem ultrapassado seus "deadlines" de reprodução.

Para um nó-folha qualquer (sistema final receptor), o grau de qualidade de visualização agregado é o próprio grau de qualidade de visualização. No nó $R_{1,2}$, o grau de qualidade de visualização agregado $\overline{Q\hat{o}S_v}_{1,1}$ é simplesmente a média aritmética dos graus de qualidade de

[2] *Cápsulas* são pacotes injetados na rede por aplicações e que contém programas para serem executados nos nós da rede. Tal conceito é intimamente ligado ao paradigma de arquiteturas de rede chamado *redes ativas* (Wetherall 1999) (Calvert et al. 1998), uma rede "inteligente"que suporta modificações dinâmicas em seu comportamento.

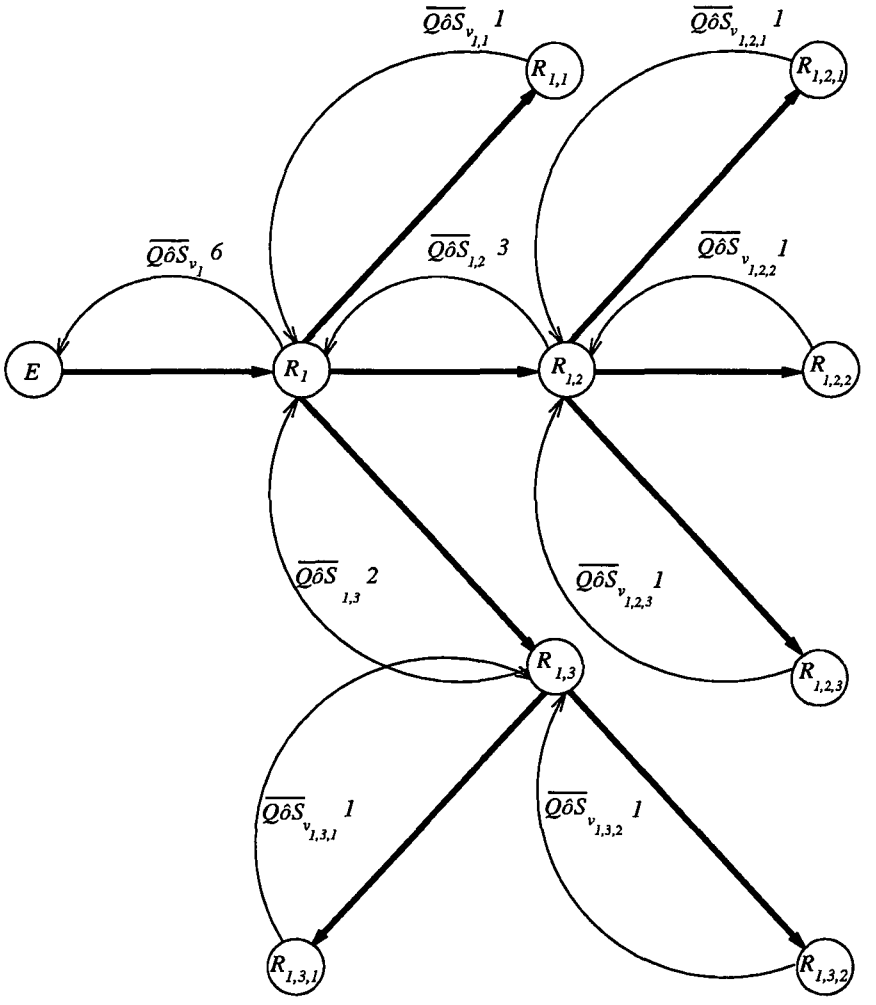

Figura 6.1: Agregação distribuída da variável de realimentação agregada.

visualização dos sistemas finais receptores $R_{1,2,1}$, $R_{1,2,2}$ e $R_{1,2,3}$; no nó $R_{1,3}$, o grau de qualidade de visualização agregado $\overline{QôS_{v1,3}}$ é a média aritmética dos graus de qualidade de visualização de $R_{1,3,1}$ e $R_{1,3,3}$. No nó R_1, o grau de qualidade de visualização agregado $\overline{QôS_{v1}}$ (que representa, no caso, o grau de qualidade de visualização agregado total $\overline{QôS_v}$) é dada por:

$$\overline{QôS_v} = \frac{\overline{QôS_{v1,2}} \times 3 + \overline{QôS_{v1,3}} \times 2 + \overline{QôS_{v1,1}}}{6}. \tag{6.1}$$

Essa estratégia distribuída de cálculo da variável de realimentação agregada é similar àquela utilizada para o cálculo da taxa explícita do serviço ABR das redes ATM.

A política de agregação distribuída descrita acima contempla melhor as heterogeneidades do SMD, por ser menos influenciada por valores extremos e por permitir que política de adaptação contemple o estado da maioria dos sistemas finais receptores, através da ponderação. Além

disso, essa estratégia minimiza a quantidade de informação de realimentação que circula na rede e reduz a possibilidade de nós receberem um fluxo além ou aquém de suas possibilidades em termos de largura de banda disponível e carga do processador. O atraso, contudo, continua sendo um aspecto crítico na abordagem com controle centralizado. Para reduzir o atraso e melhorar ainda mais a questão da justiça na adaptação, é proposto neste trabalho um *modelo de adaptação de QoS distribuído*. Tal modelo será detalhado na seção seguinte.

6.4 Modelo de Adaptação Distribuído

No modelo de adaptação de QoS distribuído, há vários CN's espalhados nos nós da rede ao invés de um único no sistema final emissor. A Figura 6.2 mostra o modelo de adaptação distribuído para seis sistemas finais receptores ($N = 6$).

No modelo de adaptação de QoS distribuído, cada nó retransmite o fluxo para os nós abaixo dele em sua sub-árvore. O nível de QoS do fluxo pode ser igual ou diferente daquele recebido por ele. Existe uma relação entre os graus de qualidade de emissão desses níveis de QoS. No exemplo, $Q\hat{o}S_E \geq Q\hat{o}S_{e_1}$; $Q\hat{o}S_{e_1} \geq Q\hat{o}S_{e_{1,2}} \wedge Q\hat{o}S_{e_1} \geq Q\hat{o}S_{e_{1,3}}$, onde $Q\hat{o}S_E$ é o grau de qualidade gerado pela aplicação no sistema final emissor, $Q\hat{o}S_{e_1}$ é o grau de qualidade do nível de QoS emitido pelo nó R_1; $Q\hat{o}S_{e_{1,2}}$ é o grau de qualidade do nível de QoS emitido pelo nó $R_{1,2}$; $Q\hat{o}S_{e_{1,3}}$ é o grau de qualidade do nível de QoS emitido pelo nó $R_{1,3}$. Todos os nós (com exceção dos nós-folhas) realimentam os nós imediatamente acima deles com o grau de qualidade recebido. O grau de qualidade de recepção de um nó pode ser no máximo igual ao grau de qualidade de recepção do nó acima dele em virtude dos pacotes descartados no "buffer" de recepção. Os nós-folhas realimentam os nós imediatamente acima deles com o grau de qualidade de visualização.

A Figura 6.3 mostra o esquema de controle considerando a topologia mostrada na figura anterior. Existem quatro mecanismos de adaptação locais, um posicionado no emissor E e outros três posicionados nos nós R_1, $R_{1,2}$ e $R_{1,3}$. Todos funcionam da mesma maneira que o mecanismo centralizado descrito na Seção 5.4.3.

O mecanismo de adaptação de nó E (sistema final emissor) é realimentado apenas por $Q\hat{o}S_{r_1}$, o grau de qualidade de recepção do nó imediatamente abaixo dele (R_1). O monitor calcula o grau de qualidade de recepção agregado $\overline{Q\hat{o}S}_{r_E}$ (nesse caso, $\overline{Q\hat{o}S}_{r_E} = Q\hat{o}S_{r_1}$) e fornece para o CN. O CN calcula o novo grau de qualidade de emissão e envia-o para o atuador

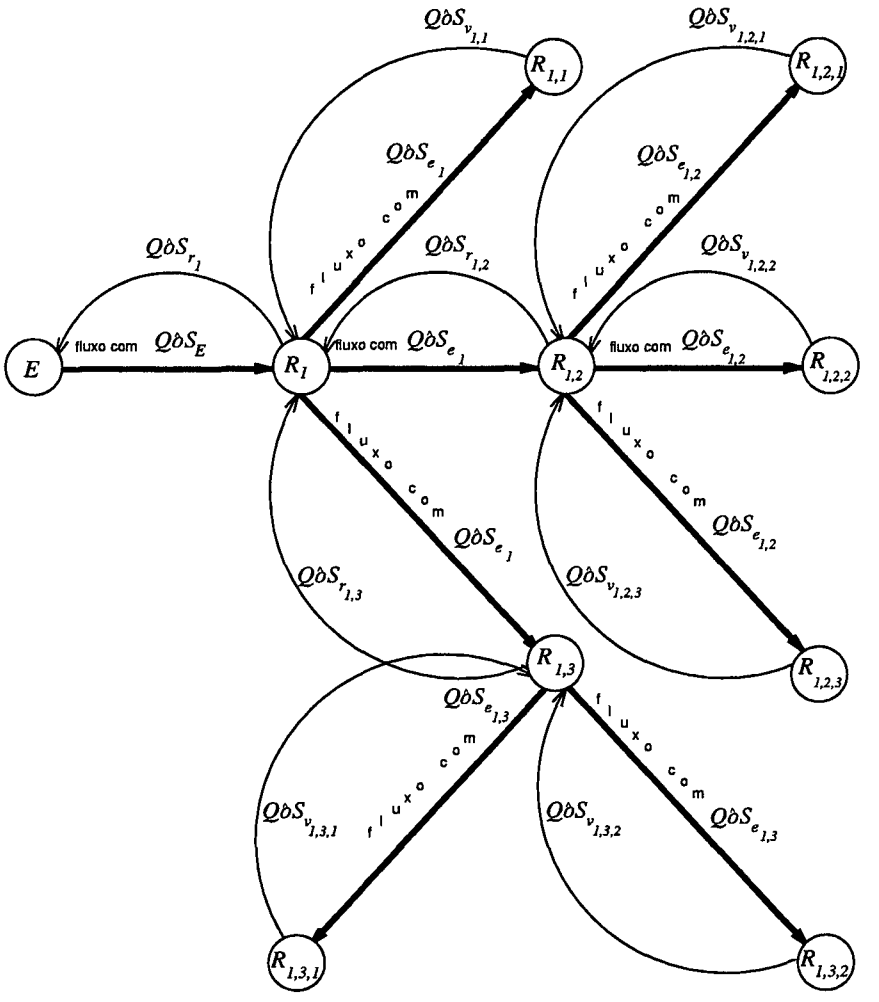

Figura 6.2: Modelo de adaptação distribuído.

do mecanismo de adaptação. O atuador usa esse grau de qualidade para selecionar em $\Omega'_{QoSLevel}$ o novo nível de QoS de emissão L'. A seguir, ele configura a aplicação para passar a emitir o fluxo multimídia com os parâmetros de QoS valorados de acordo com L'.

O mecanismo de adaptação de R_1 é realimentado pelo grau de qualidade de visualização de $R_{1,1}$ e pelos graus de qualidade de recepção de $R_{1,2}$ e $R_{1,3}$ e ($Q\hat{o}S_{v_{1,1}}$, $Q\hat{o}S_{r_{1,2}}$ e $Q\hat{o}S_{r_{1,3}}$, respectivamente). Seu monitor calcula o grau de qualidade de recepção agregado $\overline{Q\hat{o}S_{r_1}}$ e fornece-o para o CN. O CN calcula o novo grau de qualidade de emissão $Q\hat{o}S_{e_1}$ e envia-o para o atuador do mecanismo de adaptação. O atuador determina os novos valores dos parâmetros de QoS, retira os pacotes do "buffer" de recepção de pacotes, remonta-os e executa uma descompressão parcial visando alterar os valores dos parâmetros através de filtragem. Os quadros filtrados são novamente empacotados e colocados no "buffer" para que sejam enviados para os nós receptores.

O mecanismo de adaptação de $R_{1,2}$ é realimentado pelos graus de qualidade de visualização de seus três receptores: $Q\hat{o}S_{v_{1,2,1}}$, $Q\hat{o}S_{v_{1,2,2}}$ e $Q\hat{o}S_{v_{1,2,3}}$. Seu monitor calcula o grau de qualidade de recepção agregado $\overline{Q\hat{o}S}_{r_{1,2}}$ e fornece-o para o CN que calcula o novo grau de qualidade de emissão $Q\hat{o}S_{e_{1,2}}$, enviando-o para o atuador. O resto do processo ocorre de maneira similar ao anterior.

O funcionamento do mecanismo de adaptação de $R_{1,3}$ é análogo aos demais.

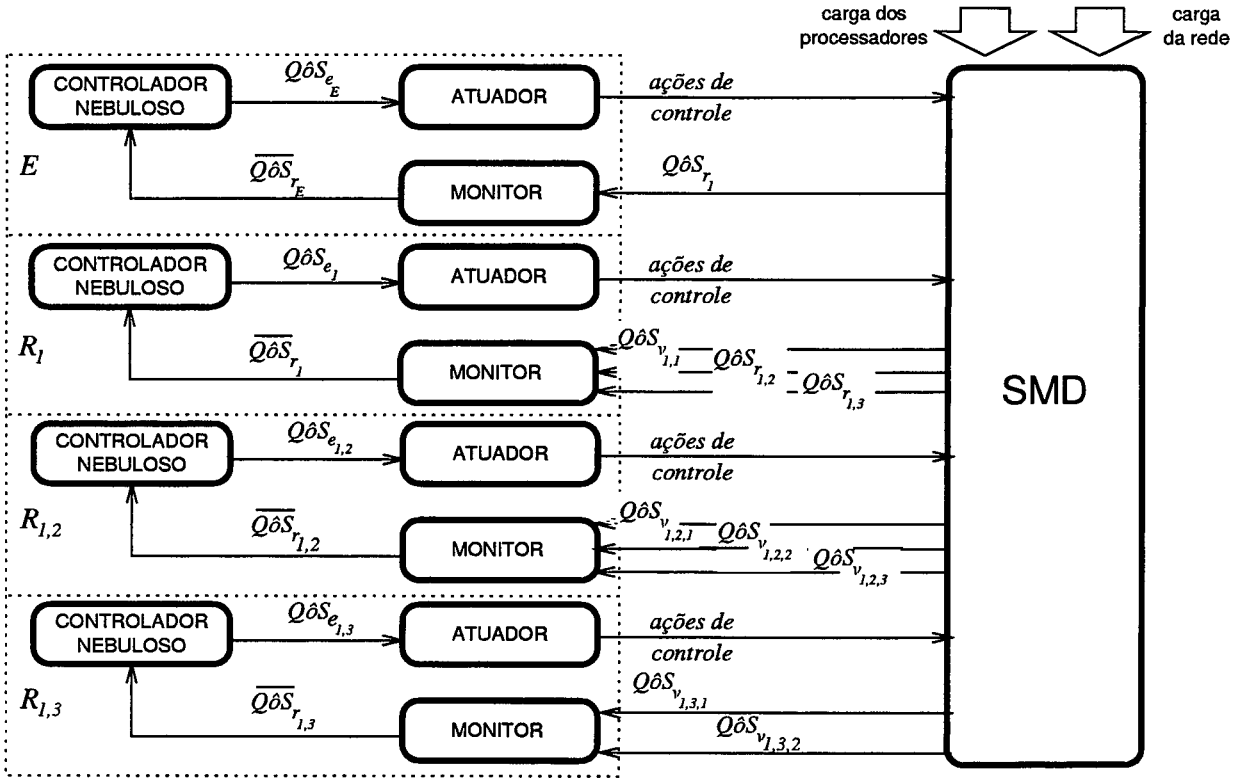

Figura 6.3: Mecanismo de adaptação de QoS distribuído.

O período T de atuação dos mecanismos de adaptação nos nós da rede deve ser igual ao período definido para o monitoramento. Como, para um dado nó R_i, os tempos de chegada da variável de realimentação fornecida pelos k nós imediatamente abaixo dele podem variar bastante, é necessário manter uma tabela, com uma entrada para cada receptor, onde são armazenados os valores mais recentes das variáveis de realimentação fornecidas por cada sistema final receptor, de maneira similar ao mecanismo de fusão de cápsulas proposto por Gonçalves et al. em (Gonçalves et al. 2000). Em cada nó R_i da árvore multiponto e a cada T segundos, o mecanismo de adaptação do nó executa a função ψ que implementa a política de agregação da variável de realimentação. Para tal, ele pega na tabela os valores de realimentação dos k nós

abaixo de R_i, calcula a informação de realimentação agregada e fornece-a para o CN.

6.5 Resumo e Discussão

Neste capítulo foi apresentada a versão preliminar da proposta de um mecanismo de adaptação de QoS distribuído. A detecção da necessidade de um modelo distribuído surgiu a partir da verificação de alguns problemas relacionados ao modelo centralizado (justiça na adaptação, escalabilidade, demora na atuação do mecanismo em virtude do atraso na recepção da variável de realimentação) quando da verificação de sua viabilidade através de experimentos.

O modelo de adaptação de QoS distribuído possui duas grandes vantagens sobre aqueles baseados em um controle centralizado no sistema final emissor. A primeira vantagem refere-se ao fato da informação de realimentação atravessar apenas uma ligação entre um nó qualquer e o nó acima dele ao invés de todas as ligações existentes entre o sistema final receptor e o sistema final emissor. Assim, em uma dado instante, haverá apenas uma mensagem de controle concorrendo pelo uso de uma ligação. No modelo de adaptação de QoS centralizado, em um dado instante, pode haver N mensagens concorrendo por uma ligação, conforme a topologia da rede, o que aumenta em muito a probabilidade de congestionamento de ligações, prejudicando a escalabilidade do modelo. Além de diminuir o tráfego e controle na rede, as informações de realimentação estarão mais atualizadas do que no esquema de controle centralizado no sistema final emissor em decorrência da menor distância a ser percorrida.

A segunda vantagem refere-se à possibilidade do controle contemplar melhor, quando da adaptação, às necessidades individuais de cada sistema final receptor. Novamente em decorrência da proximidade entre os nós e o controlador, não haverá tanta discrepância nos valores da variável de realimentação (grau de qualidade, no caso) informados por eles, fazendo com que a decisão de adaptação contemple melhor as particularidades e contexto corrente de cada sistema final receptor. Para exemplificar, seja uma situação em que os sistemas finais e os nós intermediários informam diferentes valores para os graus de qualidade de recepção de visualização, conforme a Figura 6.4.

Na Tabela 6.1 é mostrado o novo grau de qualidade de emissão usando diferentes abordagens. Por simplicidade, foi assumido que o grau de qualidade de emissão corrente em todos os nós é 1 e que o CN só usa o grau de qualidade de recepção (no caso de nós intermediários) ou visualização (no caso de sistemas finais receptores) corrente para o cálculo do erro (*i.e.*, a

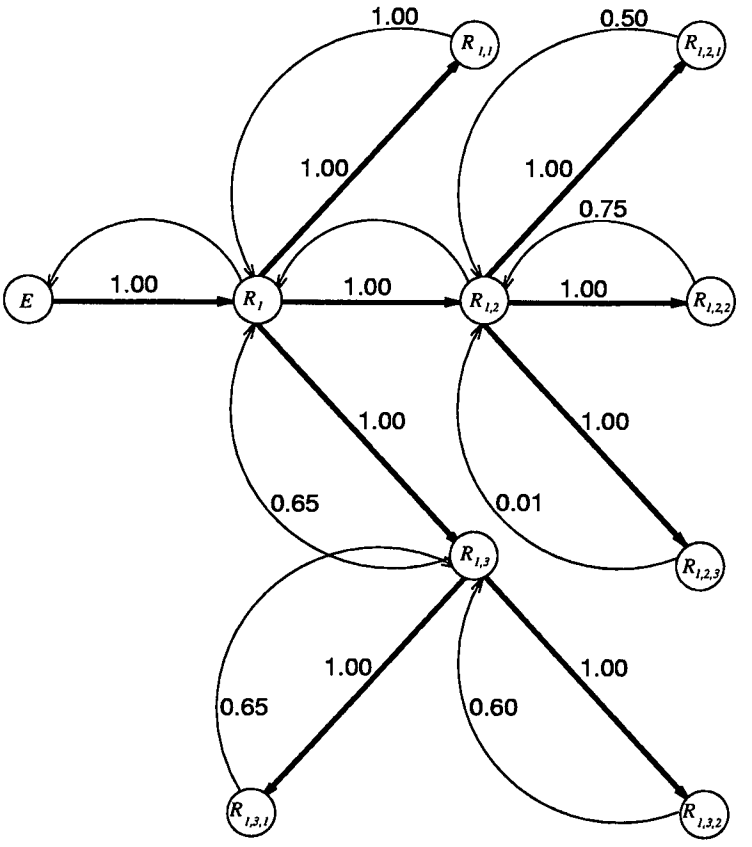

Figura 6.4: Exemplo de nós enviando diferentes valores para a variável de realimentação.

variável de realimentação não é suavizada).

No caso do modelo de adaptação de QoS centralizado, independentemente da política de agregação usada, haverá apenas um grau de qualidade de emissão, possibilitando com que determinadas ligações ou processadores dos sistemas finais receptores fiquem sobrecarregados ou sub-utilizados. No caso do modelo de adaptação distribuído, por outro lado, haverá um grau de qualidade de emissão por nó intermediário da rede cujo valor será mais próximo das disponibilidades de recursos dos nós abaixo dele.

Um problema do modelo de adaptação de QoS distribuído é relacionado ao valor de T, o período de monitoramento/atuação da adaptação. Geralmente, tal valor é definido de forma arbitrária, sem manter relação com nenhum parâmetro de QoS. No modelo centralizado, o período pode ser igual ou superior ao maior atraso existente entre um sistema final receptor e o sistema final emissor. Isso, contudo, não garante que a adaptação será direcionada para o contexto corrente de todos os sistemas finais receptores, já que, conforme a distribuição geográfica destes, os valores dos atrasos podem ser extremamente díspares fazendo com que o mecanismo trabalhe

Modelo Centralizado			
Política de Agregação	$\overline{QôS_v}$	**Erro**	**Novo** $QôS_e$
Média Aritmética	0.585	0.415	0.750
Pior Caso	0.010	0.990	0.500
Média Ponderada	0.585	0.415	0.750
Modelo Distribuído			
Emissor	$\overline{QôS_v}$ **ou** $\overline{QôS_r}$	**Erro**	**Novo** $QôS_e$
E	1.000	0.000	1.000
R_1	0.800	0.200	0.955
$R_{1,2}$	0.420	0.580	0.750
$R_{1,3}$	0.625	0.375	0.750

Tabela 6.1: Novo grau de qualidade de emissão: modelo centralizado × modelo distribuído.

com variáveis de realimentação defasadas. Assim, tanto no modelos de adaptação de QoS centralizado quanto no distribuído a determinação de um valor adequado de T exigirá, futuramente, uma análise mais aprofundada.

No modelo de adaptação de QoS distribuído, há ainda um problema adicional: a necessidade de sincronização entre os mecanismos de adaptação. Sem uma sincronização, um nó $R_{i,j}$ qualquer pode calcular um grau de qualidade de emissão $QôS_{e_{i,j}}$ enquanto o nó R_i acima dele calculou um grau de qualidade de emissão $QôS_{e_i}$ tal que $QôS_{e_i} < QôS_{e_{i,j}}$. Nesse caso, o grau de qualidade de emissão calculado em $R_{i,j}$ não poderá ser alcançado. Em uma solução simplista, o atuador de $R_{i,j}$ poderia calcular o novo nível de QoS de emissão quando $min(QôS_{e_i}, QôS_{e_{i,j}})$. Isso exige uma realimentação em duas vias e a estabilidade desta solução é uma aspecto ainda em aberto.

Outro problema do modelo de adaptação de QoS distribuído proposto é o "overhead" introduzido nos nós que aumenta o atraso fim-a-fim. Esse "overhead" é causado pelo próprio mecanismo de adaptação, principalmente em decorrência das operações de filtragem usadas para alterar o nível de QoS de emissão do nó, já que o CN possui uma complexidade computacional baixa, conforme visto no capítulo anterior. A filtragem ainda é um processo que demanda um tempo relativamente longo, mesmo considerando-se a capacidade de processamento das máquinas atuais. Assim, ela deveria ser executada via "hardware". Yeadon et al (Yeadon et al. 1996) argumentam, contudo, que mesmo que esse processo seja executado via "software", parte do atraso introduzido será compensado nos sistemas finais receptores pela diminuição do atraso representado pelo processo de descompressão, já que quanto mais degradada é a quali-

dade do fluxo multimídia, mais rápida é a descompressão. As operações de filtragem no $i^{ésimo}$ nó também podem ser evitadas quando o valor de ξ, dado pela equação abaixo, for *muito pequeno*:

$$\xi = \begin{cases} |bps(t_n) - bps(t_{n-1})| & \text{se } Q\hat{o}S_{e_i}(t_{n-1}) > Q\hat{o}S_{e_i}(t_n) \\ |Q\hat{o}S_{e_i}(t_n) - Q\hat{o}S_{e_i}(t_{n-1})| & \text{se } Q\hat{o}S_{e_i}(t_{n-1}) < Q\hat{o}S_{e_i}(t_n) \end{cases} \quad (6.2)$$

onde $Q\hat{o}S_{e_i}(t_n)$ representa o novo grau de qualidade de emissão (aquele calculado) e $Q\hat{o}S_{e_i}(t_{n-1})$ representa o grau de qualidade de emissão corrente no $i^{ésimo}$ nó, respectivamente. A primeira situação ocorre quando a decisão de adaptação consiste na redução do grau de qualidade de emissão com uma redução da taxa de bits mas a nova taxa de bits não difere muito da anterior, correspondente ao grau de qualidade calculado. Neste caso, o ganho em largura de banda não compensará o atraso introduzido pelo próprio mecanismo de adaptação (conforme visto no Capítulo 2, o atraso, além de aumentar o tempo de resposta, pode implicar também em perdas de quadros e amostras de áudio). A segunda situação ocorre quando a decisão de adaptação consiste em aumentar o grau de qualidade de emissão mas a diferença entre o calculado e o corrente é muito pequena, sendo, provavelmente, pouco percebida pelo usuário final.

Capítulo 7

CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho foi discutido um mecanismo de adaptação de QoS baseado no uso de controle nebuloso. O uso de CN's foi escolhido por parecer adequado para adaptação de QoS já que uma série de características dos sistemas multimídia distribuídos combinam com aquelas de sistemas nos quais o controle nebuloso tem sido aplicado com sucesso. Dentre essas características, destacam-se:

- não-linearidade na relação entre parâmetros de QoS, como taxa de bits e parâmetros da camada de aplicação;
- imprecisão na determinação do estado do sistema, em termos de carga da rede e de processador; e
- imprecisão do critério de avaliação do sistema (qualidade fornecida).

Além do uso de CN´s, outra característica que diferencia o mecanismo aqui proposto de outros similares é que ele é *totalmente centrado no usuário final.* Essa característica é alcançada porque o *mecanismo considera sempre a qualidade como um todo* (de acordo com a perspectiva do usuário final), alterando o valor de vários parâmetros de QoS simultaneamente (e não de um único, como na maior parte das abordagens), tendo como objetivo maximizá-la dentro do contexto do SMD através do uso da função grau de qualidade. Essa função, proposta neste trabalho, associa a taxa de utilização de recursos e as preferências do usuário às diversas combinações possíveis de valores de parâmetros de QoS da camada de aplicação.

Visando comprovar a adequação e a viabilidade do uso da proposta, foram definidos e implementados dois mecanismos de adaptação utilizando controle nebuloso. O primeiro utiliza um CN baseado em modelo de interpolação (Takagi-Sugeno) tem como variável de alimentação a taxa de perdas de pacotes e como variável de controle a taxa de bits da aplicação multimídia; o segundo utiliza um CN baseado em um modelo de clássico (Mamdani) e tem como variáveis de realimentação e controle o grau de qualidade. Os resultados dos testes realizados com os mecanismos de adaptação de QoS mostraram que o uso de CN´s pode proporcionar uma melhor identificação do estado do SMD bem como uma adaptação mais suave da qualidade sem exigir muitos recursos computacionais.

Os resultados obtidos também mostraram que uma maior disponibilidade de largura de banda não implica necessariamente em melhor qualidade se a largura de banda não for utilizada de forma adequada, através da seleção da melhor combinação de valores de parâmetros de QoS. Isso reforça a necessidade de estratégias que realizem o mapeamento taxa de bits-parâmetros de QoS-qualidade. No caso do mecanismo de adaptação de QoS aqui proposto, um atuador realiza esse mapeamento através da função grau de qualidade, usada pelo mesmo para selecionar a melhor combinação de parâmetros de QoS para a carga corrente do sistema. A eficiência do uso da função grau de qualidade pelo atuador foi comprovada na prática através de um melhor aproveitamento, sob o ponto de vista do usuário final, da largura de banda disponível para a aplicação multimídia.

No caso do segundo mecanismo, o grau de qualidade é utilizado como variável de realimentação por ser uma métrica que: (1) permite que o mecanismo de adaptação identifique o estado do SMD em termos de carga de forma tão eficiente como outros parâmetros; (2) permite que o mecanismo de adaptação identifique os reflexos da carga do SMD sobre a qualidade percebida pelo usuário final; (3) considera a natureza da aplicação; e (4) permite que a adaptação funcione tanto como uma reação à carga da rede quanto às cargas dos processadores dos sistema finais.

O primeiro aspecto é verdadeiro porque, independentemente do parâmetro de QoS usado como variável de realimentação, o mecanismo de adaptação terá apenas uma visão vaga do estado do SMD. Essa imprecisão permite que qualquer variável relacionada a algum tipo de perda possa ser usada para estimar a carga do sistema.

O segundo aspecto é verdadeiro porque o grau de qualidade informa para o mecanismo de adaptação a qualidade que o usuário final está presenciando no seu sistema final, ao contrário

de parâmetros de QoS como taxa de perdas de pacotes, taxa de ocupação de "buffers", atraso, taxa de perdas de quadros e taxa de perdas de "deadlines", que, de forma isolada, não podem ser mapeados para qualidade. Não existe ainda uma função que permita relacionar, por exemplo, o a taxa de perda de pacotes com a taxa de perda de quadros ou amostras de áudio (parâmetros de QoS mais próximos do usuário), já que esses dois parâmetros dependem de vários outros fatores. Medições que, através de um conjunto de amostras, permitam estabelecer uma relação entre perdas da camada de comunicação e perdas da camada da aplicação não podem ser generalizadas, já que são amarradas a um contexto específico em termos de algoritmos de compressão, tecnologia de rede e natureza da aplicação. A obtenção de uma função que realizasse esse mapeamento não eliminaria, ainda, o problema da qualidade estar sendo avaliada de forma unidimensional.

O terceiro aspecto é verdadeiro porque a construção da função grau de qualidade leva em consideração a natureza da aplicação ao estabelecer pesos para os parâmetros de QoS. Assim, um grau de qualidade baixo recebido pelo mecanismo de adaptação indica que um parâmetro de QoS "importante", não só sob o ponto de vista do usuário mas também da aplicação, está sendo degradado.

O último aspecto é verdadeiro porque o grau de qualidade usado como variável de realimentação é o grau de qualidade de visualização, que embute tanto as perdas ocasionadas pela rede quanto aquelas ocasionadas pelo sistema final.

Esses quatro aspectos não só validam o uso do grau de qualidade como variável de realimentação de mecanismos de adaptação de QoS como o tornam mais adequado do que outros parâmetros de QoS.

Duas dificuldades encontradas quando da implementação do mecanismo de adaptação de QoS foram o monitoramento e a atuação.

No caso do monitoramento, o uso do grau de qualidade como variável de realimentação exigiu a criação de conexões adicionais entre os sistemas finais. No mecanismo realimentado pela taxa de perdas de pacotes, isso não foi necessário já que o próprio protocolo de controle permitia o monitoramento desse parâmetro. Essa facilidade, contudo, associa o controle a parâmetros de QoS distantes do usuário final e/ou parâmetros de QoS cuja importância relativa pode ser baixa para determinadas aplicações. O atraso e a variação do atraso de rede, utilizados como variáveis de realimentação em algumas propostas, por exemplo, permitem apenas identificar de forma direta a perda em relação a um único parâmetro de QoS da camada da aplicação,

ou seja, o tempo de resposta, que não é tão importante em aplicações de dados armazenados. Tal problema poderia ser resolvido se os pacotes de controle permitissem que a aplicação também definisse informações a serem carregadas.

Quanto à atuação, as dificuldade referem-se à complexidade computacional dos filtros existentes, que pode comprometer a eficiência do mecanismo de adaptação, e à pouca exploração da escalabilidade dos algoritmos de compressão por parte dos codificadores e decodificadores, o que restringe o universo de parâmetros de QoS aptos a sofrerem adaptação.

Para minimizar os problemas relacionados à escalabilidade, justiça na adaptação e atraso, inerentes a todas abordagens de adaptação de QoS centralizadas nos sistema finais, foi também proposto neste trabalho um mecanismo de adaptação de QoS distribuído nos nós intermediários da rede. Diante das limitações do controle centralizado, especialmente em WAN's, e da provável introdução de serviços pagos na Internet, a adaptação distribuída tende, de fato, a ser mais explorada futuramente. Neste contexto, uma abordagem que parece ser bastante promissora são as redes ativas, por possibilitarem o disparo de processos nos nós da rede através de interfaces relativamente simples.

Por fim, constatou-se que o uso de mecanismos de adaptação de QoS (independentemente da abordagem usada) pode melhorar o desempenho de aplicações multimídia distribuídas tanto em termos de uso de recursos quanto qualidade oferecida. Contudo, em ambientes melhor-esforço, a qualidade pode permanecer sob um longo tempo abaixo do mínimo esperado pelos usuários. Particularmente no atual estágio da Internet, a execução de aplicações multimídia distribuídas não oferece uma qualidade mínima que permita que esse ambiente possa competir com outras formas de difusão de áudio e vídeo "tradicionais", como televisão, rádio e sistema de telefonia. Apenas o uso de abordagens recentes como o uso de serviços diferenciados/integrados, permitirá que a Internet torne-se realmente um ambiente adequado para aplicações multimídia.

A reserva de recursos, contudo, não invalidará a necessidade do uso de mecanismos de adaptação de QoS. Em virtude do custo envolvido, apenas usuários com grande disponibilidade financeira ou cuja reserva seja direcionada para a execução de aplicações QoS-rígidas, estarão dispostos a pagar por uma reserva para o pior caso (taxa de bits máxima), o que dispensa a necessidade da adaptação. No caso geral, a reserva de recursos deverá considerar o caso médio, existindo uma faixa de largura de banda (de maneira similar ao serviço ABR das redes ATM) sobre a qual a adaptação poderá atuar. Neste cenário, o uso de controle nebuloso, em virtude das vantagens já mencionadas, deverá ser mais explorado.

Devido ao fato do uso de controle nebuloso em SMD não ser ainda muito comum, há muitas questões passíveis de serem objeto de investigações futuras, entre as quais: incorporação na base de regras de outras entradas, execução de mais experimentos visando encontrar valores de parâmetros mais apropriados para o CN, avaliação do impacto das perdas de pacotes sobre a qualidade e realização de testes subjetivos para a construção de funções grau de qualidade.

Apêndice A

ALGORITMOS DE COMPRESSÃO

Este anexo fornece uma introdução às técnicas de compressão que são mais relevantes em aplicações multimídia. O objetivo não é fornecer uma visão exaustiva mas sim enfatizar aspectos relacionados a algoritmos de compressão mencionados no decorrer deste trabalho. Ela foi extraída basicamente de (Fluckiger 1995), (Gall 1991) e (Wallace 1991).

A.1 Tipos de Compressão

Conforme (Fluckiger 1995), as técnicas de compressão seguem duas estratégias:

1. *compressão sem perdas*: na compressão sem perdas, a informação original é recuperada sem qualquer alteração após o processo de descompressão, isto é, o fluxo obtido após a descompressão é exatamente idêntico àquele existente antes da descompressão. Estratégias de compressão sem perdas são exigidas por certas aplicações multimídia onde a precisão da informação é essencial, como em imagens médicas. A compressão sem perdas é também conhecida como *compressão reversível*; e

2. *compressão com perdas*: na compressão com perdas ou *compressão irreversível*, a informação obtida após a descompressão é diferente da original (antes da descompressão). Esta é a estratégia utilizada pela maior parte dos algoritmos de compressão, tanto de áudio quanto vídeo. Deve-se enfatizar, contudo, que muitas vezes as perdas ocorridas não são percebidas pelo observador.

A.2 Tipos de Compressão

As técnicas de compressão são classificadas em duas categorias principais: codificação[1] por entropia e codificação da fonte ("source encoding").

A.2.1 Codificação de Entropia

A *codificação de entropia* refere-se às técnicas de compressão que não consideram a natureza da informação a ser comprimida. Técnicas baseadas em entropia tratam todos os dados como seqüências de bits, sem tentar otimizar a compressão através do conhecimento do tipo de informação a ser comprimida, ou seja, essas técnicas ignoram a semântica da informação. Um exemplo trivial de uma codificação por entropia é a substituição uma série de 10 octetos sucessivos de valor 0 por um caracter especial - o "flag´´ - seguido do número 10.

A codificação de entropia produz uma compressão sem perdas e é geralmente executada através de duas técnicas supressão de seqüências repetitivas e codificação estatística.

Supressão de Seqüências Repetitivas

A *supressão de seqüência repetitivas* é a mais simples e antiga técnica de compressão usada em computação. Ela consiste na detecção de seqüências de bits ou octetos (de fato, caracteres) e sua substituição pelo número de ocorrências seguido do "flag". Dois octetos que são geralmente alvo da substituição são aqueles representando os caracteres 0 (em dados numéricos) e branco (em dados textuais).

Codificação Estatística

A *codificação estatística* é uma técnica de codificação de entropia mais elaborada do que a supressão de seqüências repetitivas. Ela consiste na identificação dos padrões de bits ou bytes mais freqüentes em uma dada seqüência e na sua substituição por menos bits. Os padrões menos freqüentes serão codificados com mais bits enquanto os mais freqüentes serão codificados com menos bits.

Obviamente, esta técnica implica no registro dos padrões (tanto inicial quanto a correspondente codificação) em uma tabela que é usada na compressão e descompressão. Tal tabela é

[1] O termo "codificação"("encoding") é o mais usual na terminologia de processamento de sinais digitais. Contudo, o que os algoritmos realizam é, de fato, uma compressão e não simplesmente uma codificação.

referenciada como *livro-código* ("code-book"). As duas principais formas de codificação estatística são a substituição de padrões e a codificação de Huffman.

A *substituição de padrões* é usada para a codificação de informação textual. Padrões freqüentes de caracteres são substituídos por uma única palavra. Por exemplo, a palavra "multimídia" poderia ser substituída neste texto por *M e a palavra "rede"por *R.

Na *codificação de Huffman*, para uma dada seqüência de dados, são calculadas as freqüências de ocorrências de cada octeto. As ocorrências são armazenadas em uma tabela. A partir dessa tabela, o algoritmo de Huffman determina o número mínimo de bits para representar cada caracter e atribui um código que é armazenado no livro-código. Este método é usado tanto para compressão de imagens estáticas quanto em movimento. Dependendo dos parâmetros da implementação, um novo livro-código pode ser construído para todas imagens ou para um conjunto de imagens. No caso de vídeo, o livro-código pode ser refeito para cada quadro ou para um conjunto de quadros. Em todos os casos, o sistema final onde será feita a descompressão deve receber o livro-código do sistema final onde foi feita a compressão.

A.2.2 Codificação da Fonte

A *codificação da fonte* é uma técnica de compressão dependente do sinal original. Por exemplo, um sinal de áudio tem certas características que podem ser exploradas na compressão: na fala, a supressão do silêncio é um típico exemplo de uma transformação que é estritamente dependente da semântica do sinal. De maneira similar, a pesquisa por blocos comuns entre quadros sucessivos de um fluxo de vídeo é também uma operação baseada no conhecimento da natureza do sinal.

A codificação da fonte pode produzir taxas de compressão bem mais altas do que a compressão por entropia. Porém, essas taxas estão intimamente ligadas à semântica do dado, sendo, assim, muito variáveis. Na realidade, a codificação por entropia e da fonte não são técnicas mutuamente exclusivas: na compressão de som, imagem ou vídeo, as duas técnicas são combinadas visando a obtenção da mais alta taxa de compressão possível.

A codificação da fonte pode produzir uma compressão com ou sem perdas, sendo classificada em três tipos: codificação de transformada, codificação diferencial e quantização vetorial.

Codificação de Transformada

Na *codificação de transformada*, o dado sofre uma transformação matemática de um domínio espacial ou temporal para um domínio abstrato que é mais adequado à compressão. O processo é, na maioria das vezes, reversível, isto é, aplicando a transformada inversa, o dado original é recuperado. Um exemplo de transformada é a Transformada de Fourier, que permite transformar uma medida que varia no tempo, $f(t)$, em uma função $g(\lambda)$. Essa nova função fornece a amplitude g - ou o *coeficiente* - das freqüências λ que compõem a função inicial. A função $g(\lambda)$ é a distribuição espectral de $f(t)$. Nas representações espectrais de imagens, as freqüências informam quão rapidamente as cores e a luminância mudam.

A idéia que norteia o processo de codificação de transformada é que, após a transformação, as partes mais significativas da informação - ou os coeficientes mais significativos (aqueles que contêm mais "energia") - são facilmente identificáveis e, possivelmente, agrupados em pacotes. Isso permite que os coeficientes mais significativos sejam codificados com maior precisão do que os menos significativos (de fato, alguns coeficientes podem até ser descartados). O fato da técnica de codificação de transformada considerar precisão e descartar coeficientes faz com que ela seja um processo de compressão com perda.

Além da Transformada de Fourier, há também a Transformada de Hadamar, Transformada de Haar e Transformada de Karhunen Loeve; a transformada matemática geralmente usada para imagens é Transformada de Cosseno Discreta ("Discrete Cosine Transform" - DCT).

Codificação Diferencial

O princípio da *codificação diferencial* ou *codificação preditiva* é codificar apenas a diferença entre o valor real de uma amostra e o próximo valor previsto. Essa diferença é chamada *diferença de predição* ou *termo de erro.*

A codificação diferencial é particularmente adequada para sinais nos quais valores sucessivos são significativamente diferentes de zero mas não diferem muito uns dos outros, como no caso dos sinais de vídeo. Os três principais esquemas de codificação diferencial são:

1. modulação de código de pulso diferencial;
2. modulação delta; e
3. modulação de código de pulso diferencial adaptável;

A *modulação de código de pulso diferencial* ("differential pulse code modulation" - DPCM) é um esquema onde o processo de predição (realizado através de uma função que calcula o próximo valor) não varia no tempo. O caso mais simples consiste na transmissão no tempo t_n da diferença entre o valor da amostra em t_n (o valor real) e o valor da amostra em t_{n+1} (o valor previsto).

A *modulação delta* é um caso particular da codificação DPCM, no qual a diferença entre o valor previsto e o valor corrente é codificado com apenas um bit, indicando que o valor do sinal será incrementado ou decrementado em um "quantum" (uma constante pré-definida). A modulação delta é adequada para codificar sinais cujos valores não mudam muito rapidamente para uma dada freqüência de amostras, isto é, é um esquema de codificação adequado para sinais de baixa freqüência.

A *modulação de código de pulso diferencial adaptável* ("adaptive differential pulse code modulation" - ADPCM) é uma versão mais sofisticada da codificação DPCM que, ao invés de usar uma função de predição fixa, usa uma função variável para estimar características de curta duração do sinal amostrado. Assim, uma extrapolação adaptável é aplicada. Como no DPCM, apenas o termo do erro é transmitido.

Quantização Vetorial

A *quantização vetorial* é um caso especial de substituição de padrão no qual o fluxo de dados é dividido em blocos chamados vetores. No caso de uma imagem, por exemplo, um vetor é geralmente um pequeno bloco, retangular ou quadrado, de pixels. O livro-código contém padrões de vetores (pré-definidos ou dinamicamente montados). Para cada vetor de uma amostra, o livro-código é consultado para verificar qual padrão (vetor do livro-código) que melhor combina com o vetor da amostra. Apenas a referência desse padrão (o número de sua entrada na tabela) é transmitida.

Para evitar distorções resultantes de uma diferença significativa entre o dado real e o padrão, junto com a referência é transmitido o termo de erro.

A.3 Compressão de Imagem

Existe um grande número de algoritmos de compressão para vídeo. Nesta seção serão vistos apenas dois deles cujo funcionamento é similar a vários outros. São eles os algoritmos JPEG e

MPEG-1.

A.3.1 O Padrão JPEG

O padrão JPEG é um padrão ISO originário do "Joint Photographic Expert Group"da ISO/IEC JTC1/Subcomitê 2 (Wallace 1991). Ele foi desenvolvido em colaboração com a ITU.

O JPEG é um padrão de compressão para imagens coloridas ou com níveis de cinza. Para a compressão, ele usa uma combinação de DCT, quantização, supressão de seqüências repetitivas e codificação de Huffman, permitindo os seguintes modos de operação:

1. codificação seqüencial: é realizada uma única varredura na imagem, da esquerda para a direita, do topo para a base. Esse modo de operação é com perdas;

2. codificação progressiva: a codificação é feita através de múltiplas varreduras na imagem. Esse modo de operação também é com perdas;

3. codificação sem perdas: o processo de compressão é reversível; e

4. codificação hierárquica: a codificação contempla vários níveis de resolução que podem ser descomprimidos separadamente.

Passos da Codificação Progressiva

Para ilustrar como o algoritmo de compressão JPEG usa as técnicas de codificação vistas nas seções anteriores, serão vistos os passos usados por ele para a codificação progressiva, mostrados na Figura A.1.

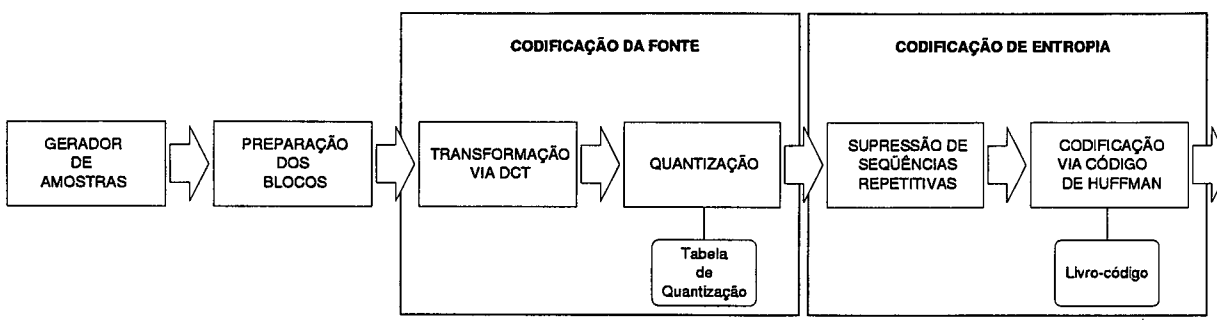

Figura A.1: Passos para compressão de imagens usando o algoritmo JPEG com o modo de operação seqüencial

O primeiro passo é a *preparação dos blocos*, na qual a imagem é dividida em blocos de 8×8 pixels. Considerando-se, por exemplo, uma imagem de 640×480 pixels representada por três componentes: a luminância Y e as diferenças de cores U e V. Se a relação entre esses componentes é 4:1:1, então o componente Y consiste de uma matriz 640×480 e os outros dois consistem de matrizes 320×240. A preparação dos blocos irá fornecer para o passo seguinte 4800 blocos para o componente Y, 1200 para o U e 1200 para o V.

O segundo passo consiste na transformação dos blocos usando DCT. A submissão dos blocos à transformação ocorre componente por componente e, dentro de um componente, da esquerda para a direita, do topo para a base, em um esquema chamado de ordenamento não-entrelaçado. Os blocos são compostos de 64 valores que representam a amplitude do sinal amostrado que é função de duas coordenadas espaciais, ou seja, $a = f(x, y)$ onde x e y são as duas dimensões. Após a transformação, obtém-se a função $c = g(F_x, F_y)$ onde c é um coeficiente e F_x e F_y são as freqüências espaciais para cada direção. O resultado é outro bloco de 64 valores onde cada valor representa um coeficiente DCT - isto é, uma determinada freqüência - e não mais a amplitude do sinal na posição amostrada (x, y). O coeficiente $g(0, 0)$, correspondente às freqüências zero, é chamado de coeficiente DC. Ele representa o valor médio das 64 amostras. Como em um bloco representando uma porção da imagem os valores amostrados geralmente variam pouco de um ponto para outro, os coeficientes de mais baixa freqüência serão altos e os de média e alta freqüência terão valores baixos ou zero, podendo ser descartados. A energia do sinal é concentrada nas freqüências espaciais mais baixas. A Figura A.2 (Fluckiger 1995) é uma representação tridimensional da transformação DCT.

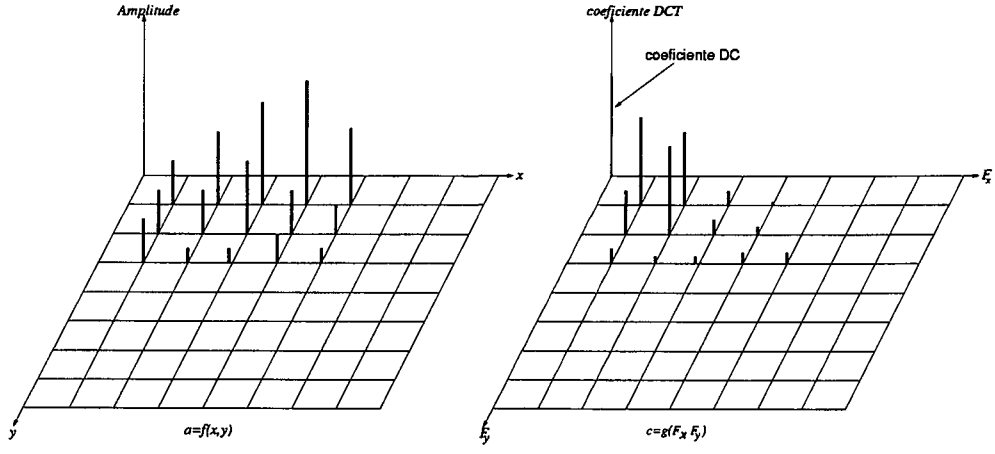

Figura A.2: Representação tridimensional da transformação DCT: antes da transformação (esquerda); depois da transformação (direita)

Em uma imagem, os coeficientes de média e baixa freqüência ocorrerão quando há uma mudança brusca (em um desenho preto-e-branco, a mudança de uma zona totalmente branca para um zona com uma linha preta representando parte da figura, por exemplo). Em fotografias, o tipo de imagem-alvo do JPEG, as transições entre as zonas da imagem são suaves.

O terceiro passo é a quantização. Nesse passo, são introduzidas perdas (até o passo da transformação, o processo era totalmente reversível[2]). O passo da quantização consiste em normalizar cada coeficiente DCT através de sua divisão por valores pré-definidos, armazenados em uma tabela chamada *tabela de quantização*. Cada elemento da tabela pode ter um valor de 1 a 255. A tabela determina quais coeficientes serão mantidos ou descartados e quais serão representados com mais ou menos precisão (se todos os elementos da tabela têm valor 1, a quantização não terá nenhum efeito). O incremento dos valores dos coeficientes aumenta a taxa de compressão e reduz a fidelidade da imagem resultante. A seqüência de coeficientes DC de cada bloco ($g(0,0)$) também é codificada usando DPCM, o que significa calcular o termo de erro existente entre os coeficientes DC de blocos adjacentes.

O último passo antes da transmissão ou armazenamento consiste na aplicação de algum esquema de codificação de entropia. Nesse passo, o algoritmo JPEG aplica ou a codificação de Huffman ou alguma técnica mais dinâmica. A ordem com que os coeficientes são pegos é em ziguezague, visando maximizar a probabilidade de ocorrência de valores idênticos sucessivos.

A Figura A.3 mostra a um exemplo de codificação usando o modo de codificação seqüencial.

O algoritmo JPEG foi concebido para compressão de imagens estáticas mas ele pode ser usado também para compressão de vídeo, sendo referenciado como M-JPEG ("motion"-JPEG). Os resultados são bons em termos da qualidade da imagem, mas a largura de banda requerida é alta (entre 8 e 10 Mbps).

A.3.2 O Padrão MPEG

O padrão MPEG é, na verdade, uma família de padrões para gravação e transmissão de informações de áudio e vídeo digitais. O primeiro da série foi o MPEG-1, publicado sob a referência ISO 11172 (Gall 1991). O grupo ISO tem realizado as especificações do padrão MPEG em fases distintas, onde cada fase tem como alvo uma aplicação específica. À cada fase, foi dado um nome: MPEG1, MPEG-2, MPEG-3 e MPEG-4. O padrão MPEG-1 tem como

[2]Na prática, é difícil encontrar codificadores que calculem DCT e IDCT (a transformação inversa) com uma precisão que assegure que nenhuma diferença ocorrerá.

alvo aplicações de áudio-vídeo armazenados em CD-ROM com resolução SIF ("Standard Interchange Fomat" - resolução média), exigindo uma largura de banda em torno de 1.2 MBps; o padrão MPEG-2 tem como alvo imagens com qualidade de TV e múltiplos canais de áudio com qualidade de CD, exigindo uma largura de banda de 4 a 6 MBps; o padrão MPEG-3 tinha como alvo imagens com qualidade HDTV, sendo abandonado a partir do momento que o padrão MPEG-2 passou a englobar esse tipo de aplicação; o padrão MPEG-4 foi concebido para videoconferência, utilizando pouca largura de banda.

Nesta seção, será analisado apenas o funcionamento do padrão MPEG-1, um padrão MPEG otimizado para obter taxas de compressão de até 26:1.

O algoritmo utilizado pelo MPEG-1, além de usar a correlação espacial (como o JPEG), faz uso da correlação temporal entre os quadros para fazer a compressão. Essa correlação é explorada através da divisão dos quadros em três tipos, como será visto a seguir.

Quadros de Referência e Intracodificados

A idéia que norteia a exploração da correlação temporal é que em uma seqüência de quadros uma boa parte da informação é comum a eles, ou seja, os quadros possuem áreas semelhantes ou mesmo iguais que podem ser codificadas apenas uma vez. Assim, determinados quadros comprimidos armazenam apenas diferenças em relação a outros quadros. Um quadro que contém informações necessárias para a reconstrução de um ou mais quadros é chamado de *quadro de referência.*

Sejam três quadros de um vídeo, como na Figura A.4 (a). Como pode ser visto, os quadros possuem áreas comuns, isto é, áreas de igual conteúdo, como aquelas marcadas nos quadros na Figura A.4 (b). Tais áreas, contudo, estão situadas em diferentes posições nos três quadros. Essa diferença de posição é representada através de um vetor chamado *vetor de movimento* (Figura A.4 (c)) e os blocos nos quais esse vetor será aplicado são chamados *blocos combinantes* ("matching blocks"). O tamanho desses blocos depende dos componentes da imagem. No MPEG-1, uma imagem é formada por três componentes ou planos: um plano para luminância e dois planos que representam a diferença de cor que são sub-amostrados. Assim, um bloco combinante é, na prática, um quadrado de 16×16 pixels no plano da luminância e quadrados de 8×8 pixels para cada um dos planos que representam a diferença de cor. A combinação desses três quadrados é chamada de *macrobloco*[3].

[3] O termo "macrobloco"não deve ser confundido com os blocos de 8×8 pixels usados no JPEG (e também no

Supondo que o quadro 3 tenha macroblocos em comum com o quadro 1. Supondo, ainda, que o quadro 3 é construído a partir do quadro 1 (e somente dele). Neste caso, o quadro 3 é um *quadro predito* ("predicted frame") ou quadro P. Ele é construído a partir do quadro de referência 1 que passa a ser um *quadro intracodificado* ("intracoded frame") ou quadro I. Supondo ainda que o quadro 2 tem macroblocos em comum com o quadro 1 e o quadro 3 (Figura A.4 (d)). Assim, conceitualmente, o quadro 2 pode ser reconstruído usando pedaços dos quadros 1 e 3, desde que o quadro 3 esteja disponível quando o quadro 2 é codificado. Isso implica que os três quadros têm que ser temporariamente armazenados. O quadro 2 é chamado de um *quadro bidirecional* ou quadro B, sendo construído a partir da interpolação do intraquadro 1 e do quadro predito 3.

Muitas vezes dois macroblocos não combinam totalmente. Neste caso, existe uma diferença representada aritmeticamente (o erro do termo). As áreas de um quadro P ou B para os quais não há nenhum bloco combinante são codificadas como os macroblocos dos quadros I.

Existem algumas seqüências-padrão para quadros I, P e B: $IBBBPBBBI$, $IBBPBBPBBI$ e $IBBPBBPBBPBBI$. Quanto mais quadros B tem a seqüência, maior será taxa de compressão obtida, porém, às custas de uma diminuição a correlação temporal entre eles e entre os quadros de referência, prejudicando, assim, a qualidade da imagem. Além disso, os quadros I servem como pontos de sincronização, sendo estimado que o atraso máximo entre as ocorrências de dois quadros desse tipo não deve exceder 300 ou 400 milissegundos. Em aplicações de reprodução de vídeo onde são oferecidas operações VCR, o intervalo de ocorrência entre quadros de referência (I ou P) não deve exceder 150 milissegundos.

Compressão de quadros I

Os quadros do tipo I são comprimidos de maneira muito semelhante à compressão dos quadros JPEG no modo seqüencial. Cada plano de luminância e diferença de cor é dividido em blocos de 8×8 pixels que são transformados em domínios de freqüência usando DCT. O passo de quantização é aplicado usando a tabela de quantização. Como resultado, certos coeficientes geralmente serão descartados. As séries de coeficientes mais significativos de cada bloco (coeficientes DC) são codificadas usando a técnica DPCM (apenas a diferença entre dois valores DC é codificada). Os coeficientes de cada bloco são ordenados em ziguezague e um supressor de seqüências repetitivas é aplicado. Finalmente, é aplicada a codificação de Huffman.

MPEG-1) para eliminar redundâncias espaciais via DCT.

Compressão de quadros P e B

Na compressão de quadros do tipo P e B, para cada macrobloco, é pesquisado no quadro de referência o melhor macrobloco combinante. A diferença entre o macrobloco real e o melhor macrobloco combinante é calculada na forma de um vetor de movimento. O termo de erro (que também é um macrobloco) é transformado via DCT. Os passos seguintes são a quantização, o ordenamento em ziguezague, a supressão de seqüências repetitivas e a aplicação da codificação de Huffman. Os coeficientes DC são codificados do mesmo modo que os demais, ao contrário do que ocorrer no algoritmo JPEG e nos quadros do tipo I. O vetor de movimento de cada bloco é codificado usando a técnica DPCM já que os vetores de movimento adjacente não são significativamente diferentes. A seqüência resultante é submetida à codificação de Huffman.

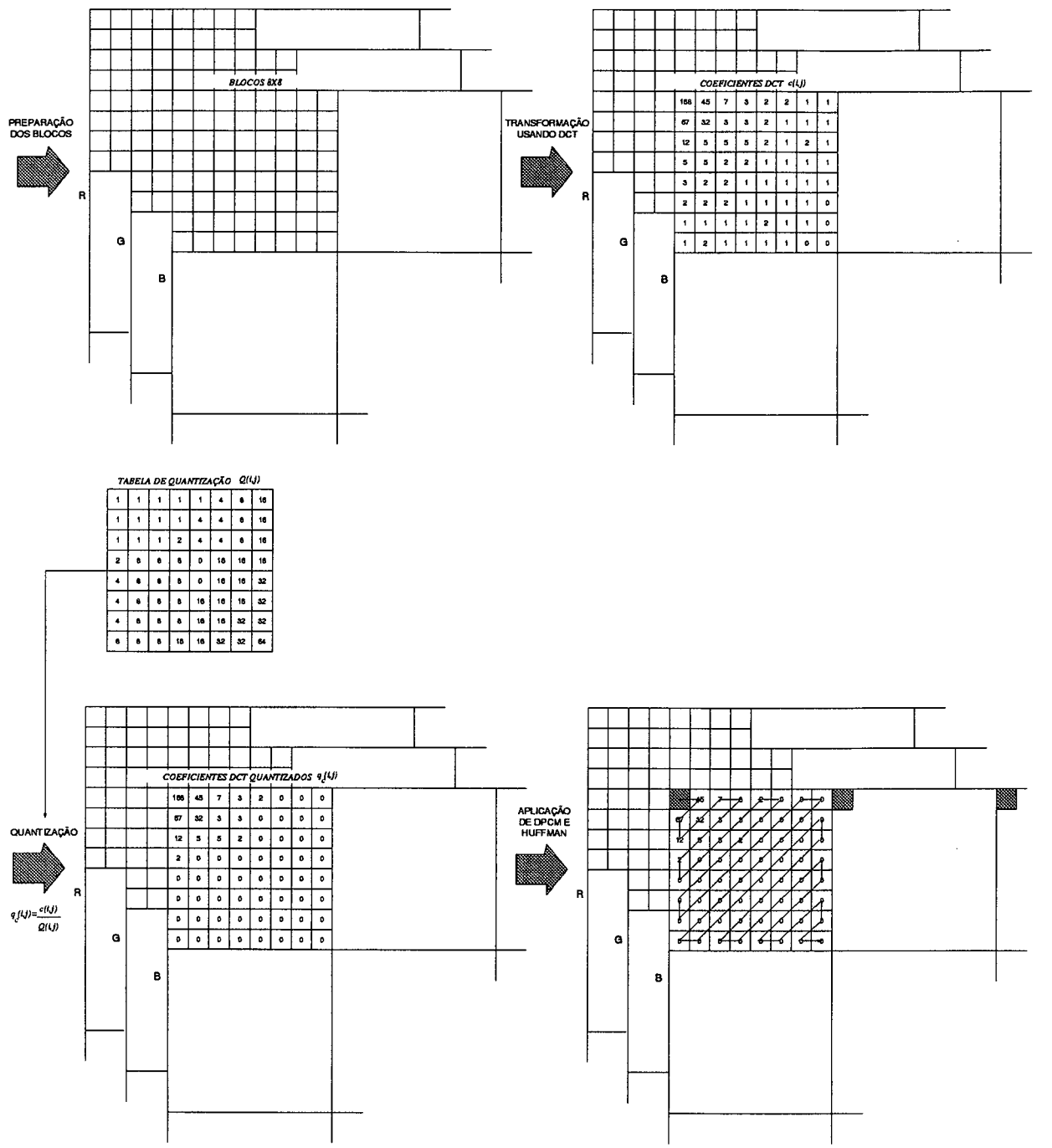

Figura A.3: Exemplo de compressão usando o algoritmo JPEG

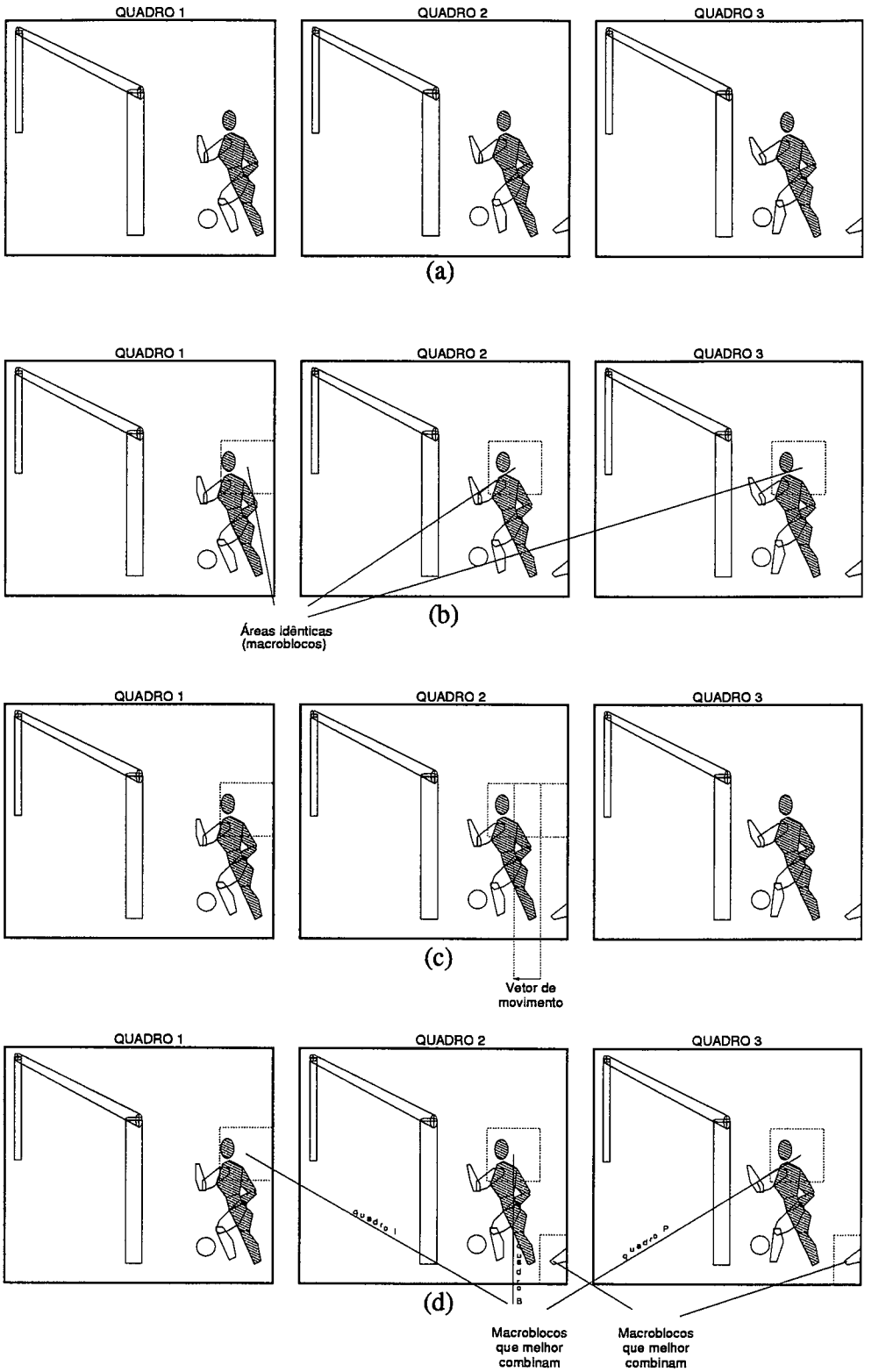

Figura A.4: Exploração da correlação temporal usando o algoritmo MPEG-1

Apêndice B

CONTROLADORES NEBULOSOS

Este anexo fornece uma introdução à teoria que embasa o controle nebuloso, tendo sido extraído de (Sandri e Correa 1999).

B.1 Conjuntos Nebulosos

A Teoria dos Conjuntos Nebulosos foi introduzida por Lotfi Zadeh em (Zadeh 1965). Segundo essa teoria, um conjunto nebuloso A do universo de discurso X é definido por uma função de pertinência

$$\mu_A : X \mapsto [0, 1].$$

Essa função μ_A ("membership function" ou *função de pertinência*) associa a cada elemento $x \in X$ um grau de compatibilidade com o conceito expresso por A:
se $\mu_A(x) = 1$, x é completamente compatível com A;
se $\mu_A(x) = 0$, x é completamente incompatível com A;
se $0 < \mu_A(x) < 1$, x é parcialmente compatível com A, com um grau $\mu_A(x)$.

B.2 Controladores Nebulosos

As técnicas de controle nebuloso originaram-se com as pesquisas e projetos de E. H. Mamdani (Mamdani e Baaklini 1975) e ganharam espaço como área de estudo em diversas instituições de ensino, pesquisa e desenvolvimento do mundo, sendo até hoje uma importante aplicação da Teoria dos Conjuntos Nebulosos.

Ao contrário dos controladores convencionais em que o algoritmo de controle é descrito analiticamente por equações algébricas ou diferenciais, através de um modelo matemático, um controlador nebuloso utiliza-se de regras lógicas no algoritmo de controle, com a intenção de descrever numa rotina a experiência humana, intuição e heurística para controlar um processo (Zadeh 1965).

Uma *variável lingüística* pode ser definida por uma quádrupla

$$< X, \Omega, \mathcal{T}(X), M >,$$

onde x é o nome da variável, Ω é o universo de discurso de x, $\mathcal{T}(x)$ é um conjunto de nomes para valores de x, e M é uma função que associa um grau de pertinência a cada elemento de $\mathcal{T}(X)$.

A Figura B.1 ilustra a variável lingüística *velocidade* com os termos nebulosos dados por

$$\{Negativa\ Alta,\ Negativa\ Baixa,\ Zero,\ Positiva\ Alta,\ Positiva\ Baixa\}.$$

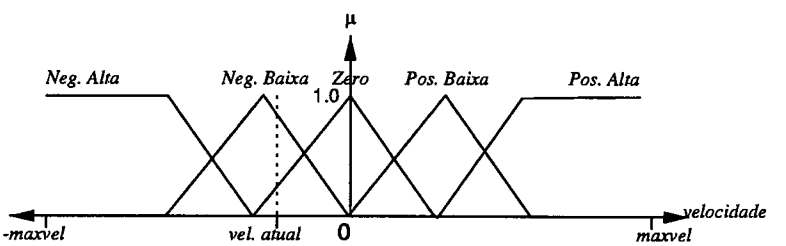

Figura B.1: Termos lingüísticos que mapeiam a variável *Velocidade*

A Figura B.2 é a representação da estrutura básica de um controlador nebuloso, como descrito em (Lee 1990a). Muitas variações são propostas na literatura de acordo com o objetivo do projeto, mas esse é um modelo geral o suficiente para a identificação dos módulos que o compõem, fornecendo uma idéia do fluxo da informação.

A *interface de codificação* faz a identificação dos valores das variáveis de entrada, as quais caracterizam o estado do sistema (variáveis de estado), e as normaliza em um universo de discurso padronizado. Estes valores são então codificados, com a transformação da entrada "crisp" (não-nebulosa) em conjuntos nebulosos para que possam se tornar instâncias de variáveis lingüísticas.

A *base de conhecimento* consiste de uma base de dados e uma base de regras, de maneira a

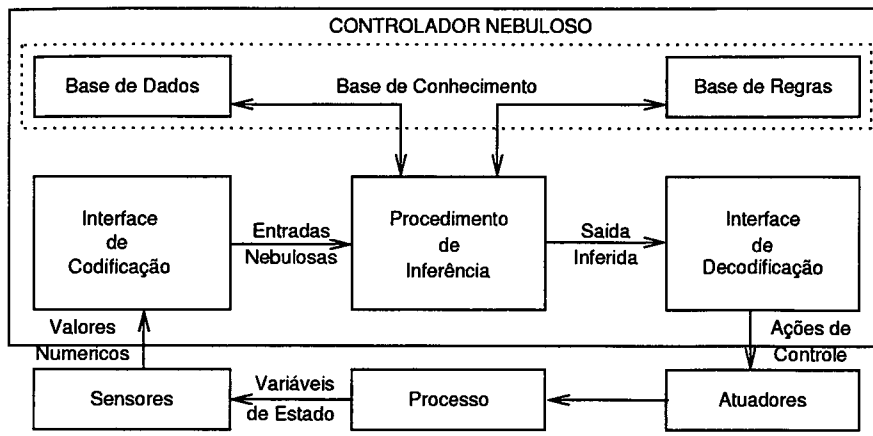

Figura B.2: Estrutura de um controlador Nebuloso

caracterizar a estratégia de controle e as suas metas.

Na *base de dados* ficam armazenadas as definições sobre discretização e normalização dos universos de discurso, e as definições das funções de pertinência dos ter-mos nebulosos.

A *base de regras* é formada por estruturas do tipo

se *premissas* então *conclusão*

Por exemplo:

se *Erro é Negativo Grande* e Δ*Erro é Positivo Pequeno* então *Velocidade é Positiva Pequena*

Estas regras, juntamente com os dados de entrada, são processados pelo *procedimento de inferência*, o qual infere as ações de controle de acordo com o estado do sistema.

Em um dado controlador nebuloso, é importante que existam tantas regras quantas forem necessárias para mapear totalmente as combinações dos termos das variáveis, isto é, que a base seja completa, garantindo que exista sempre ao menos uma regra a ser disparada para qualquer entrada. Também são essenciais a consistência (ausência de contradições) e a interação entre as regras, gerenciada pela função de implicação de modo a contornar as situações de ciclo.

As premissas são relacionadas pelos conectivos lógicos, dados pelo operador de conjunção (*e*) e o operador de disjunção (*ou*). Em geral as regras tem a forma de um sistema de múltiplas entradas e múltiplas saídas (MIMO), mas que pode ser transformado em vários sistemas com múltiplas entradas e uma saída (MISO). Por exemplo, a regra MIMO

se x_1 é A_1 e ... e x_n é A_n então y_1 é C_1 e ... e y_m é C_m

é equivalente a m regras MISO

$$\text{se } x_1 \text{ é } A_1 \text{ e } ... \text{ e } x_n \text{ é } A_n \text{ então } y_j \text{ é } C_j \ .$$

Em geral, não se aceitam conectivos *ou* na conclusão.

Um controlador nebuloso é um sistema especialista simplificado onde a conseqüência de uma regra não é aplicada como antecedente de outra. Assim, o processo de inferência consiste em:

1. verificação do grau de compatibilidade entre os fatos e as cláusulas nas premissas das regras;
2. determinação do grau de compatibilidade global da premissa de cada regra;
3. determinação do valor da conclusão, em função do grau de compatibilidade da regra com os dados e a ação de controle constante na conclusão (precisa ou não); e
4. agregação dos valores obtidos como conclusão nas várias regras, obtendo-se uma ação de controle global.

Os tipos de controladores nebulosos encontrados na literatura são os modelos clássicos, compreendendo o modelo de Mamdani e o de Larsen, e os modelos de interpolação, compreendendo o modelo de Takagi-Sugeno e o de Tsukamoto (Lee 1990a) (Lee 1990b).

Os modelos diferem quanto à forma de representação dos termos na premissa, quanto à representação das ações de controle e quanto aos operadores utilizados para implementação do controlador.

B.2.1 Modelo Clássico de Controle Nebuloso

Nos modelos clássicos de controle nebuloso, a conclusão de cada regra especifica um termo nebuloso dentre um conjunto fixo de termos (geralmente em número menor que o número de regras). Estes termos são usualmente conjuntos nebulosos convexos como triângulos, funções em forma de sino ("bell-shaped") e trapézios.

Dado um conjunto de valores para as variáveis de estado, o sistema obtém um conjunto nebuloso (muitas vezes subnormalizado), como o valor da variável de controle. Este conjunto nebuloso representa uma ordenação no conjunto de ações de controle aceitáveis naquele momento. Finalmente, uma ação de controle global é selecionada dentre aquelas aceitáveis em um processo conhecido como decodificação.

Sejam as regras R_j codificadas como:

$$\text{se } x_1 \text{ é } A_{1,j} \text{ e } ... \text{ e } x_n \text{ é } A_{n,j} \text{ então } y_j \text{ é } C_j \ .$$

No modelo clássico, o processamento de inferência é feito da seguinte maneira:

- seja x_i uma variável de estado, definida no universo X_i, a realização de x_i é definida como o valor $x_i^* \in X_i$ que esta assume em X_i em um dado momento;
- a *compatibilidade* da $i^{\text{ésima}}$ premissa da $j^{\text{ésima}}$ regra com x_i^*, $1 \leq i \leq n$, com $A_{i,j}$ da regra R_j, $1 \leq j \leq m$, é definida por

$$\alpha_{i,j} = \mu_{A_{i,j}}(x_i^*), 1 \leq i \leq n, 1 \leq j \leq m \tag{B.1}$$

- com as premissas de uma dada regra avaliadas, a *compatibilidade global* α_j da regra R_j, $1 \leq j \leq m$, com os x_i^* é determinada através da função $\top$

$$\alpha_j = \top(\alpha_{1,j}, ..., \alpha_{n,j}), 1 \leq j \leq m \tag{B.2}$$

- o α_j assim obtido é relacionado com o respectivo conjunto nebuloso C_j do conseqüente da regra R_j, dando origem a um conjunto C_j', $1 \leq j \leq m$, através do operador de implicação I

$$\mu_{C_j'}(y) = I(\alpha_j, \mu_{C_j}(y)), \forall\, y \in Y \tag{B.3}$$

- um operador ∇ faz a *agregação* das contribuições das várias regras acionadas C_j' num único conjunto nebuloso C'

$$\mu_{C_j}(y) = \nabla(\mu_{C_1'}(y), ..., \mu_{C_m'}(y)), \forall\, y \in Y \tag{B.4}$$

Os modelos clássicos seguem estritamente os passos mostrados acima, sendo que no modelo de Mamdani $\top(a,b) = min(a,b)$, $I = min(a,b)$ e $\nabla(a,b) = max(a,b)$ e no modelo de Larsen $\top(a,b) = a * b$, $I = a * b$ e $\nabla(a,b) = max(a,b)$.

A Figura B.3 (parte superior) ilustra o processo de raciocínio do modelo de Mamdani.

Os controladores de Mamdani e Larsen necessitam da utilização de uma interface de decodificação para gerar a ação de controle, isto é, escolher um único valor no suporte de C'.

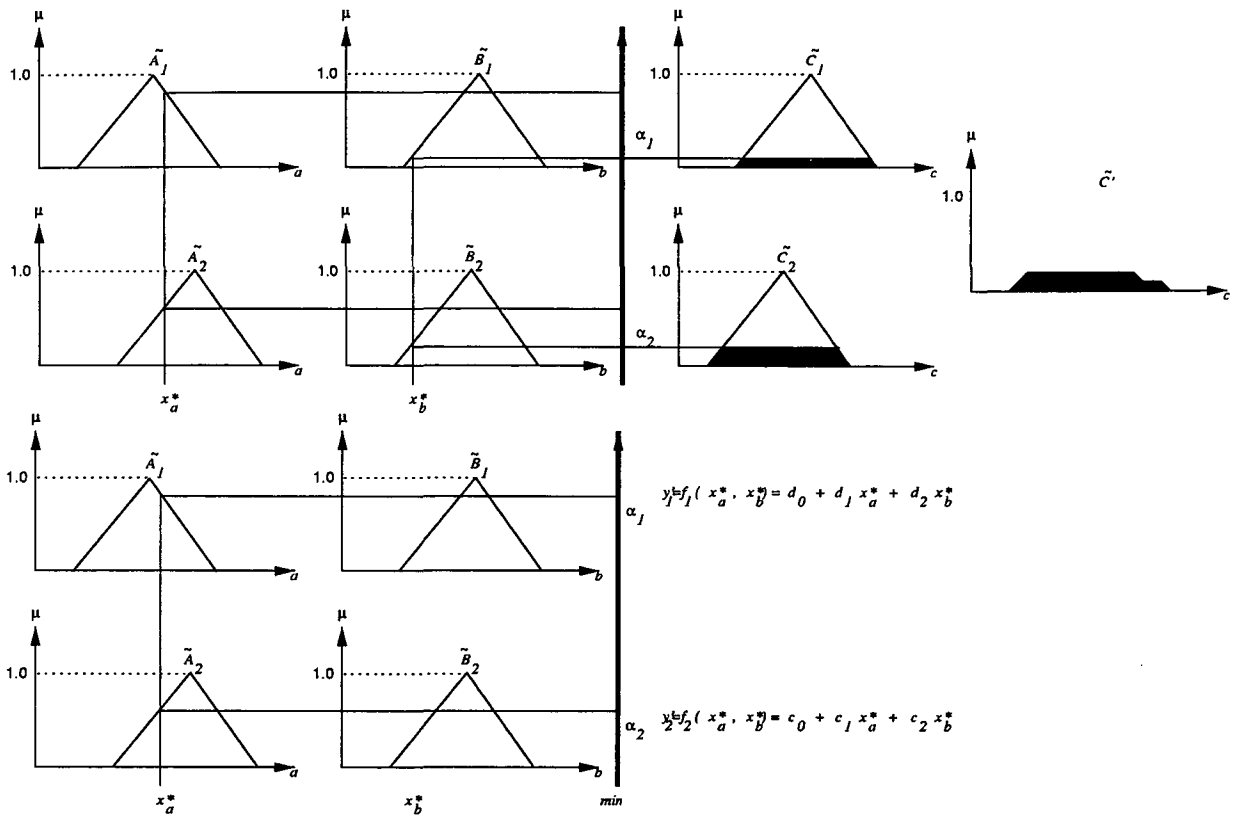

Figura B.3: Modelos de Mamdani e Takagi-Sugeno

B.2.2 Modelo de Interpolação

Nos modelos de interpolação, cada conclusão é dada através de uma função estritamente monotônica, usualmente diferente para cada regra. No modelo de Takagi-Sugeno, a função é uma combinação linear das entradas, tendo como parâmetros um conjunto de constantes. No esquema de Tsukamoto, a função é geralmente não-linear, tendo como domínio os possíveis graus de compatibilidade entre cada premissa e as entradas.

Em ambos os esquemas, obtém-se, para cada regra, um único valor para a variável de controle. Finalmente, uma ação de controle global é obtida fazendo-se uma média ponderada dos valores individuais obtidos, onde cada peso é o próprio grau de compatibilidade entre a da regra e as entradas, normalizado.

Para os modelos de interpolação, também são válidos os passos 1, 2 e 3 descritos nos modelos clássicos. No entanto, a operação de implicação do passo 4 determina uma ação de controle precisa para cada regra. Essas ações individuais são interpoladas no passo 5, gerando ação de controle única e precisa.

O modelo de Tsukamoto exige que pelo menos os conjuntos nebulosos C_j, que estão as-

sociados com os conseqüentes das regras, funções monotônicas. No passo 4, o método de interpolação obtém um valor preciso y'_0 relativo à ação de controle da regra R_j, que é dado por:

$$y'_j = \mu_{C_j}(\alpha_j). \tag{B.5}$$

Por sua vez, no passo 5, os valores obtidos como conclusão nas várias regras são agregados em uma única ação de controle precisa y', através de uma média ponderada, dada por:

$$y' = \frac{\sum_{j=1}^{n}(\alpha_j.y'_j)}{\sum_{j=1}^{n}\alpha_j} \tag{B.6}$$

Neste caso, a interface de decodificação não é utilizada.

O modelo de Takagi e Sugeno exige que todos os termos nebulosos $A_{i,j}$ sejam funções monotônicas e que as conclusões das regras sejam dadas por funções:

$$f_j(x_1, ..., x_m) = d_{0,j} + d_{1,j}.x1 + ... + d_{m,j}.x_m \tag{B.7}$$

onde cada d_k é uma constante.

A ação de controle obtida por cada regra R_j é dada por:

$$y'_j = f_j(x_1^*, x_2^*, ..., x_m^*) \tag{B.8}$$

A ação de controle y' é então obtida pela Equação B.6, como no modelo de Tsukamoto. Na Figura B.3 (parte inferior), ilustra-se a inferência através do método de Takagi-Sugeno de duas regras MISO.

É interessante notar que um controlador nebuloso de Takagi-Sugeno se comporta como um sistema do tipo PD quando existe somente uma única regra na base, dada por

$$\text{se } x_1 = erro \text{ e } x_2 = A_2 \text{ então } c = d_0 + d_1.x_1 + d_2.x_2$$

onde $x_1 = erro$, $x2 = \Delta erro$, $d_0 = 0$ e os termos A_i são tais que $\mu_{A_i}(x) = 1, \forall X \in X_i$.

Quando $x_1 = erro$, $x_2 = \Delta erro$ e $d_{0,j} = 0, 1 \leq j \leq m$, um controlador do tipo Takagi-Sugeno se comporta como se os resultados de um conjunto de controladores PD, cada um definido para uma região do espaço de estados, fossem interpolados.

B.2.3 Passos para Construção de um Controlador Nebuloso

De uma maneira geral, pode-se descrever as tarefas de construção de um controlador nebuloso brevemente como:

1. definição do modelo e das características operacionais, para estabelecer as particularidades da arquitetura do sistema (como sensores e atuadores) e definição das propriedades operacionais do controlador nebuloso do projeto, como o tipo de controlador, operadores a serem utilizados, interface de decodificação etc.;

2. definição dos termos nebulosos de cada variável. Para garantir suavidade e estabilidade deve-se permitir que haja uma sobreposição parcial entre conjuntos nebulosos vizinhos; e

3. definição do comportamento do controle, que envolve a descrição das regras que atrelam as variáveis de entrada às propriedades de saída do modelo.

No projeto de controladores nebulosos é necessária, portanto, a definição de alguns parâmetros, obtidos a partir da experiência do projetista ou através de experimentos. Para um determinado processo, alguns destes parâmetros são fixos - os parâmetros estruturais -, e outros - os parâmetros de sintonização - variam com o tempo: São parâmetros estruturais:

- número de variáveis de entrada e saída;
- variáveis lingüísticas;
- funções de pertinência parametrizadas;
- intervalos de discretização e normalização;
- estrutura da base de regras;
- conjunto básico de regras; e
- recursos de operação sobre os dados de entrada;

São parâmetros de sintonização:

- universo de discurso das variáveis;
- parâmetros das funções de pertinência (por exemplo, núcleo e suporte).

Propriedades da base de regras como a completude, consistência, interação e robustez precisam ser testadas. A robustez relaciona-se com a sensibilidade do controle frente a ruídos ou algum comportamento incomum não-modelado. Para ser medida, introduz-se um ruído aleatório de média e variância conhecidas e observa-se então a alteração dos valores das variáveis de saída.

A sintonização é uma tarefa complexa devido à flexibilidade que decorre da existência de muitos parâmetros, exigindo esforço do projetista na obtenção do melhor desempenho do controlador. Alguns dos parâmetros podem ser alterados por mecanismos automáticos de adaptação e aprendizado, contudo, normalmente é tarefa o projetista o treinamento e a sintonia da maioria dos parâmetros. Esta sintonização é feita através de busca, uma atividade típica em Inteligência Artificial.

A sintonização pode ser feita da seguinte maneira:

1. desenvolve-se um controlador simples, que simule um controlador proporcional com:
   - conjunto de variáveis mais relevantes;
   - baixo número de variáveis lingüísticas.
2. incrementa-se o conhecimento conforme a experiência resultante do processo:
   - buscando-se novas variáveis lingüísticas ou físicas para contornar as dificuldades;
   - ajustando-se as funções de pertinência e os parâmetros do controlador;
   - adicionando-se regras ou modificando a estrutura de controle.
3. valida-se a coerência do conhecimento incorporado com novas condições de operação para o sistema.

Essas tarefas necessitam de plataformas sofisticadas, com interfaces poderosas e que permitam uma rápida inferência. Isto é proporcionado pelos pacotes integrados, dedicados a análise de modelos nebulosos (Gomide et al. 1995).

B.2.4 Aprendizado Usando Redes Neurais

Nos últimos anos, tem havido um crescente interesse sobre os chamados sistemas "neuro-fuzzy" (Wang e Mendel 1992) (Jang 1993). A rigor, qualquer sistema que misture os paradigmas de sistemas nebulosos e sistemas conexionistas poderia ser chamado de "neuro-fuzzy"', como, por

exemplo, a utilização de um controlador nebuloso para alterar dinamicamente a taxa de aprendizado de uma rede neural. No entanto, o termo é utilizado para um tipo específico de sistema que de certa forma engloba os dois paradigmas. Nestes sistemas, os termos e regras de um sistema nebuloso são aprendidos mediante a apresentação de pares (entrada, saída desejada). Eles apresentam dois comportamentos distintos, dependendo de estar numa fase de aprendizado ou numa fase de processamento da informação: na fase de aprendizado, eles têm um comportamento de redes neurais, e na fase de processamento, eles se comportam como um sistema nebuloso. Estes sistemas são capazes de solucionar problemas apresentados pelos paradigmas em que se baseiam.

Referências Bibliográficas

Abdelzaher, T. e Shin, K. G. (1998). End-host Architecture for QoS-Adaptative Communication, *IEEE 4th Real-Time Technology and Applications Symposium*, Denver, Colorado, USA.

Aldrifge, R., Davidoff, J., Ghanbari, M., Hands, D. e Pearson, D. (1995). Measurement of Scene-dependent Quality Variations in Digitally Coded Television Pictures, *IEEE Proc. Vis. Image Signal Process* **142**(3): 149–154.

Amir, E., McCanne, S. e Zhang, H. (1995). An Application Level Video Gateway, *3st ACM International Conference on Multimedia (Multimedia'95)*, San Francisco, USA.

Baiceanu, V., Cowan, C., McNamee, D., Pu, C. e Walpole, J. (1996). VBR MPEG Video Coding with Dynamic Bandwidth Renegotiation, *Workshop on Resource Allocation Problems in Multimedia Systems*, Washington, DC, USA.

Bezdek, J. C. (1981). *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press.

Bocheck, P., Campbell, A. T., Chang, S.-F. e Liao, R. R.-F. (1999). Scalable Feedback Control for Multicast Video Distribution in the Internet, *9th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'99)*, Basking Ridge, USA.

Bogatinovski, M., Trajkovski, G. e Spasenovski, B. (1998). Fuzzy Controller for Video Conference Traffic in B-ISDN, *IEEE 6th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD'98)*, São Paulo, Brazil, p. 55–59.

Bolot, J.-C. e Turletti, T. (1998). Experiences with Control Mechanisms for Packet Video in The Internet, *Computer Comm. Review, ACM SIGComm* **28**(1): 4–16.

Bolot, J.-C., Turletti, T. e Wakeman, I. (1994). Scalable Feedback Control for Multicast Video Distribution in the Internet, *ACM SIGComm'94*, p. 58–67.

Braden, R., Clark, D. e Shenker, S. (1994). Integrated Services in the Internet Architecture: an Overview, *Relatório Técnico RFC 1633*, The Internet Society.

Busse, I., Deffner, B. e Schulzrinne, H. (1995). Dynamic QoS of Multimedia Applications Based on RTP, *1st International Workshop on High Speed Networks and Open Distributed Platforms*, St. Petesburg, Russia.

Calvert, K. L., Bhattacharjee, S. e Sterbenz, J. (1998). Directions in Active Networks, *IEEE Communications Magazine* **36**(10): 72–78.

Campbell, A. T., Coulson, G. e Hutchison, D. (1998). Transporting QoS Adaptive Flows, *Multimedia Systems* **6**(3): 167–178.

Chatterjee, S. e Strosnider, J. (1995). Distributed Pipeline Scheduling: End-To-End Analysis of Heterogeneous, Multi-Resource Real-Time Systems, *15th IEEE International Conference on Distributed Computing Systems*.

Correa, C. (1999). *Uso de Controladores Genéticos na Construção de Controlador Nebuloso para o Controle de Altitude de um Satélite Artificial Durante a Fase de Apontamento*, Master's thesis, Instituto Nacional de Pesquisas Espaciais, São José dos Campos, Brasil.

Delgrossi, L., Halstrick, C., Hehmann, D., Herrtwich, R. G., Krone, O., Sandvoss, J. e Vogt, C. (1993). Media Scaling for Audiovisual Communication with the Heildelberg Transport System, *1st ACM International Conference on Multimedia (Multimedia'93)*, Anaheim, Anaheim, USA, p. 99–104.

Eleftheriadis, A. e Anastassiou, D. (1995). Meeting Arbitrary QoS Constraints Using Dynamic Rate Shaping of Coded Digital Video, *5th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'95)*, Durham, New Hampshire, p. 95–106.

Fischer, S., Hafid, A., Bochmann, G. V. e de Meer, H. (1997). Cooperative QoS Management in Multimedia Applications, *IEEE 4th Int. Conf. on Multimedia Computing and Systems (ICMCS'97)*, Ottawa, Canada, p. 303–310.

Fischer, S., Salem, M.-V. e Bochmann, G. V. (1997). Application Design for Cooperative QoS Management, *IFIP 5th International Workshop on Quality of Service (IWQoS'97)*, New York, NY, p. 191–194.

Fluckiger, F. (1995). *Understanding Networked Multimedia: Applications and Technology*, Prentice-Hall.

Fry, M., Seneviratne, A., Vogel, A. e Witana, V. (1996). Delivering QoS Controlled Media on the World Wide Web, *IFIP 4th International Workshop on Quality of Service (IWQoS'96)*, Paris.

Fukuda, K., Wakamiya, N., Murata, M. e Miyahara, H. (1998). On Flow Aggregation for Multicast Video Transport, *IFIP 6th International Workshop on Quality of Service (IWQoS'98)*, Napa Valley, CA, USA, p. 13–22.

Gall, D. L. (1991). MPEG: A Video Compression Standard for Multimedia Applications, *Communication of the ACM* **34**(4).

Gecsei, J. (1997). Adaptation in Distributed Multimedia Systems, *IEEE Multimedia* p. 58–95.

Gomide, F. A. C., Gudwin, R. R. e Tanscheit, R. (1995). Conceitos Fundamentais da Teoria dos Conjuntos Nebulosos, *6th International Fuzzy Systems Association World Congress - IFSA'95*, p. 1–38.

Gonçalves, P. A. S., Rezende, J. F. e Duarte, O. C. M. B. (2000). An Active Service for Multicast Video Distribution, *Journal of the Brazilian Computer Society* **7**(1): 43–51.

Hafid, A., von Bochmann, G. e Dssouli, R. (1998). Distributed Multimedia Application and QoS: a Review, *Eletronic Journal on Networks and Distributed Processing* **2**(5): 1–50.

Hull, D., Feng, W. e Liu, J. W. S. (1995). Enhancing The Performance and Dependability of Real-Time Systems, *IEEE International Computer Performance and Dependability Symposium*, Erlangen, Germany, p. 174–182.

Int (1996). *Methods for Subjective Determination of Transmission Quality*. Recommendation ITU-T P.800.

Int (2000). *Methodology for Subjective Assessment of the Quality of Television Pictures*. Recommendation ITU-R BT.500-7.

ISO (1984). *Open Systems Interconnection: Transport Service Definition.* International Standard 8072.

Jacobson, V. (1988). Congestion Avoidance and Control, *ACM SIGComm'88*, p. 314–329.

Jang, J.-S. R. (1993). ANFIS: Adaptive-Network-Based Fuzzy Inference Systems, *IEEE Transactions on Systems, Man and Cybernetics* **23**(3): 665–685.

Jones, B. L. e McManus, P. R. (1996). Graphic Scaling of Qualitative Terms, *SMPTE Journal* p. 1166–1171.

Jones, C. e Atkinson, D. J. (1998). Development of Opinion-Based Audiovisual Quality Models for Desktop Video-Teleconferencing, *IFIP 6th International Workshop on Quality of Service (IWQoS'98)*, Napa, CA USA, p. 18–25.

Kawachiya, K., Ogata, M., Nobuhiko, N. e Tokuda, H. (1995). QoS Control of Continuos Media Architecture and System Support, *Relatório Técnico RT0108*, IBM.

Knightly, E. W. e Zhang, H. (1996). Connection Admission Control for RED-VBR, A Renegotiation-Based Service, *IFIP 4th International Workshop on Quality of Service (IWQoS'96)*, Paris.

Koliver, C. e Farines, J.-M. (2001). Um Controlador Nebuloso para Adaptação de QoS, *XIX Simpósio Brasileiro de Redes de Computadores(SBRC'2001)*, Florianópolis, Brazil, p. 33–49.

Koliver, C., Farines, J.-M. e Fraga, J. S. (2000 b). Controle Dinâmico de QoS Baseado no Uso do Protocolo RTCP e Lógica Difusa, *VI Simpósio Brasileiro de Multimídia e Sistemas Hipermídia (SBMidia'2000)*, Natal, Brazil.

Koliver, C., Farines, J.-M., Fraga, J. S. e Reis, H. L. (2000 a). Um Modelo para Adaptacão de QoS Orientado ao Usuário Final, *XVIII Simpósio Brasileiro de Redes de Computadores(SBRC'2000)*, Belo Horizonte, Brazil, p. 135–150.

Koliver, C., Farines, J.-M., Fraga, J. S. e Sandri, S. (1999). Uma Abordagem para Adaptação de QoS em Aplicações Multimídia Distribuídas, *V Simpósio Brasileiro de Multimídia e Sistemas Hipermídia (SBMidia'99)*, Goiânia, Brazil, p. 213–232.

Koliver, C., Nahrstedt, K. O., Farines, J.-M., Fraga, J. S. e Sandri, S. (2001). Specification, Mapping and Control for QoS Adaptation, *Relatório Técnico DAS2000-01*, Federal University of Santa Catarina, Brazil. (submitted to The Journal of Real-Time Systems.

Koren, G. e Shasha, D. (1995). Skip-Over: Alghoritms and Complexity for Overloaded Systems That Allow Skips, *IEEE 16th Real-Time System Symposium (RTSS'95)*, Pisa, Italy, p. 11o–117.

Krasic, C. e Walpole, J. (1999). QoS Scalability for Streamed Media Delivery, *Relatório Técnico CSE-99-011*, Oregon Graduate Institute os Science and Technology, Oregon, USA.

Lakshman, V., Misshra, P. P. e Ramakrishnan, K. K. (1997). Transporting Compressed Video over ATM networks with Explicit Rate feedback Control, *IEEE INFOCOM'97*, Kobe, Japan, p. 38–47.

Lee, C. C. (1990a). Fuzzy Logic in Control Systems: Fuzzy Logic Controller, *IEEE Transactions on Systems, Man and Cybernetics* **20**(2): 404–418.

Lee, C. C. (1990b). Fuzzy Logic in Control Systems: Fuzzy Logic Controller, *IEEE Transactions on Systems, Man and Cybernetics* **20**(2): 419–430.

Li, B. e Nahrstedt, K. (1998). An Open Task Control Model for Quality of Service Adaptation, *14th International Conference of Advanced Science and Technology (ICAST 98)*, Naperville, USA, p. 29–41.

Li, B. e Nahrstedt, K. (1999). A Control-based Middleware Framework for Quality of Service Adaptation, *IEEE Journal on Selected Areas in Communications (JSAC)* **17**(9): 1632–1650.

Li, B., Xu, D., Nahrstedt, K. e Liu, J. W. S. (1998). End-to-End QoS Support for Adaptive Applications over the Internet, *SPIE International Symposium on Voice, Video and Data Communication*, p. 1–5.

Mamdani, E. H. e Baaklini, N. (1975). Prescriptive Method for Deriving Control Policy in a Fuzzy Logic Controller, *Eletronic Letters* **11**: 625–626.

Maruchek, M. J. e Strosnider, J. K. (1995). An Evaluation of the Graceful Degradation Properties of Real- Time Schedulers, *21th Annual International Symposium on Fault-Tolerant Computing*, Pasadena, California.

McCanne, S., Jacobson, V. e Vetterli, M. (1996). Receiver-Driven Layered Multicast, *Computer Communications Review* **26**(4): 117–130.

McGurk, H. e MacDonald, J. W. (1976). Hearing Lips and Seeing Voices, *Nature* **26**: 746–748.

Mercer, C. W., Rajkumar, R. e Tokuda, H. (1993). Applying Hard Real-Time Technology to Multimedia Systems, *Workshop of the Role of Real-Time in Multimedia/Interactive Computing Systems*, Raleigh-Durham, NC.

Mercer, C. W. e Tokuda, H. (1994). Processor Capacity Reserves: Operating System Support for Multimedia Applications, *IEEE 1st Int. Conf. on Multimedia Computing and Systems (ICMCS'94)*.

Nagarajan, R. (1993). *Quality of Service Issues in High Speed Networks*, PhD thesis, University of Masschusetts, Masschusetts.

Nahrstedt, K. O., Hossain, A. e Kang, S.-M. (1996). A Probe-Based Algorithm for QoS Specification and Adaptation, *IFIP 4th International Workshop on Quality of Service (IWQoS'96)*, Paris.

Nahrstedt, K. O. e Steinmetz, R. (1995). Resource Management in Networked Multimedia Systems, *IEEE Computer* **28**(5): 52–63.

Nakajima, J. e et al. (1991). Multimedia/Real-Time Extensions for the MACH Operating System, *Usenix Summer Conference*, Nashville, Tennessee, USA, p. 183–198.

N.G. Duffield, K. R. . A. R. (1998). SAVE: an Algorithm for Smoothed Adaptive Video over Explicit Rate Networks, *IEEE Transactions on Networking* **6**(6): 717–728.

Nichols, K., Blake, S., Baker, F. e Black, D. (1998). Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers, *Relatório Técnico RFC 2474*, The Internet Society.

Ortega, A. e Ramchandran, K. (1995). Forward-Adaptive Quantization with Optimal Overhead Cost for Image and Video Coding with Applications to MPEG Video Coders, *ST/SPIE Digital Video Compression '95*, San Jose, California.

Ott, M., Reininger, D. e Luo, W. (1996). Adaptive and Scalable QoS for Multimedia Using Hierarchical Contracts, *4th ACM International Conference on Multimedia (Multimedia'96)*, Boston, USA, p. 399–400.

RAC (n.d.). *General Aspects of Quality of Service and System Performance in IBC.* RACE D510.

Ramanathan, P. (1997). Graceful Degradation in Real-Time Control Applications Using (m,k)-Firm Guarantee, *27th Annual International Symposium on Fault-Tolerant Computing (FTCS '97)*, Seattle, USA, p. 132–141.

Reininger, D., Raychaudhuri, D. e Ott, M. (1998). A Dynamic Quality of Service Framework for Video in Broadband Networks, *IEEE Network* **12**(6): 22–45.

Reis, H. L. (2000). *Implementação de um Mecanismo de Adaptação de Qualidade de Serviço para uma Aplicação de Vídeoconferência*, PhD thesis, Universidade Federal de Santa Catarina, Florianopolis, Brazil.

Richards, A., Rogers, G., Antoniades, M. e Witana, V. (1998). Mapping User Level QoS from a Single Parameter, *2nd International Conference on Multimedia (MMNS'98)*.

Rowe, L. A. e Smith, B. C. (1992). A Continuous Media Player, *3th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV'92)*, San Diego, USA.

Sandri, S. A. e Correa, C. (1999). Lógica Nebulosa, *V Escola de Redes Neurais*, Rio de Janeiro, Brazil, p. c073–c090.

Schulzrinne, H., Casner, S., Frederick, R. e Jacobson, V. (1996). RTP: a Transport Protocol for Real-Time Applications, *Relatório Técnico RFC 1889*, GMD Fokus, Berlin, German.

Silveira, R. M. e Ruggiero, W. V. (2000). Sistema de Ajuste dos Coeficientes de Quantização MPEG em Tempo-Real para Vídeo sob Demanda, *SBC XVIII Simpósio Brasileiro de Redes de Computadores(SBRC'2000)*, Belo Horizonte, Brazil.

Sisalem, D. (1998). Fairness of Adaptive Multimedia Applications, *IEEE International Conference on Communications (ICC'98)*, Atlanta, USA.

Sisalem, D. e Schulzrinne, H. (2000). The Direct Adjustment Algorithm: a TCP-Friendly Adaptation Scheme, *1st International Workshop Quality of Future Internet Services (QofIS'2000)*, Berlin, Germany.

Steinmetz, R. (1995). Analyzing the Multimedia Operating System, *IEEE Multimedia* **2**(1): 145–158.

T. Nakajima, H. T. (1994). A Continuous Media Application Supporting Dynamic QoS Control on Real-Time MACH, *2st ACM International Conference on Multimedia (Multimedia'94)*, Anaheim, California, p. 289–297.

Verscheure, O., Garcia, X., Karlsson, G. e Hubaux, J.-P. (1998). User-Oriented QoS in Packet Video Delivery, *IEEE Network* **12**(6): 12–21.

Vogel, A., Kerhervé, B., von Bochmann, G. e Gecsei, J. (1995). Distributed Multimedia Applications and Quality of Service: a Survey, *IEEE MultiMedia* **2**(2): 10–19.

Waldeegg, D. B. (1997). A CPU Scheduling Model for Respecting Multimedia Temporal QoS in General Purpose Operating Systems, *Relatório técnico*, Telecom Bretagne, England.

Wallace, G. K. (1991). The JPEG Still-Picture Compression Standard, *Communications of the ACM* **34**(4): 30–44.

Wang, L. e Mendel, J. M. (1992). Generating Fuzzy Rules by Learning from Examples, *IEEE Transactions on Systems, Man and Cybernetics* **22**(6): 1414–1427.

Watson, A. e Sasse, M. A. (1996). Evaluating Audio and Video Quality in Low-Cost Multimedia Conferencing Systems, *Interacting with Computers* **8**(3): 255–275.

Watson, A. e Sasse, M. A. (1998). Measuring Perceived Quality of Speech and Video in Multimedia Conferencing Applications, *6st ACM International Conference on Multimedia (Multimedia'98)*, Bristol, England, p. 55–60.

Welling, G., Michelitsch, G., Ott, M. e Reininger, D. (1996). Dynamic Bandwidth Allocation for Distributed Multimedia with Adaptive QoS, *Workshop on Resource Allocation Problems in Multimedia Systems*, Washington, DC, USA.

Wetherall, D. (1999). Active Network Vision and Reality: Lessons from a Capsule-Based System, *Operating System Review* **34**(14): 64–79.

Winkler, S. (1999). A Perceptual Distortion Metric for Digital Color Video, *SPIE Human Vision and Electronic Imaging*, San Jose, USA.

Yeadon, N., Garcia, F., Hutchinson, D. e Mauthe, A. (1996). Filters QoS Support Mechanisms for Multipeer Communications, *IEEE Journal on Selected Areas in Communications (JSAC)* **14**(7): 1245–1262.

Zadeh, L. A. (1965). Fuzzy Sets, *Information & Control* (8): 338–353.

Zhang, H. e Knightly, E. W. (1995). RED-VBR: A New Approach To Support VBR Video in Packet-Switching Networks, *IEEE 6th Workshop on Network and Operating System Support for Digital Audio and Video*, Durham, NH, p. 275–286.

Zhang, L., Deering, S. e Estrin, D. (1993). RSVP: a New Resource ReSerVation Protocol, *IEEE network* **7**(5): 8–?