

UM CONTROLADOR NEBULOSO PARA ADAPTAÇÃO DE QoS

Cristian Koliver^{1*}

Jean-Marie Farines²

¹Universidade de Caxias do Sul
Departamento de Informática
C.P. 1352, Caxias do Sul, RS, Brasil
95001-970
ckoliver@ucs.tche.br

² Universidade Federal de Santa Catarina
Departamento de Automação e Sistemas
Florianópolis, SC, Brasil 88040-900
farines@lcmi.ufsc.br

ABSTRACT

In this paper, we describe the implementation of a fuzzy controller which adjusts dynamically the bit rate of distributed multimedia applications. The controller, deployed in the sender's side, uses the packet loss rate of the receivers - which is obtained from RR packets of RTCP protocol - as feedback and computes the new bit rate by means of linear functions, in agreement with the Takagi-Sugeno's fuzzy controller model. In this work we also describe an approach to map the bit rate to application QoS parameters in order to allow the controller to adapt the application with regard to the computed bit rate.

Key-words: distributed multimedia systems, quality of service, protocol RTP, adaptation, fuzzy controller

RESUMO

Neste trabalho, descrevemos a implementação de um controlador nebuloso ("fuzzy") dedicado ao ajuste dinâmico da taxa de bits para aplicações multimídia distribuídas. O controlador nebuloso, posicionado no lado do emissor, usa a taxa da perdas de pacotes dos receptores - que é obtida através dos pacotes RR do protocolo RTCP - como variável de realimentação e calcula a nova taxa de bits através de funções lineares, de acordo com o modelo Takagi-Sugeno de controladores nebulosos. Neste artigo, também descrevemos uma abordagem para realizar o mapeamento da taxa de bits para parâmetros de QoS da camada de aplicação a fim de ajustar o fluxo multimídia à taxa de bits calculada.

key-words sistemas multimídia distribuídos, qualidade de serviço, protocolo RTP, adaptação, controlador nebuloso.

*Autor financiado pela CAPES e CNPq, em doutoramento na UFSC.

1 Introdução

A qualidade das aplicações multimídia é expressada freqüentemente através de atributos chamados *parâmetros de qualidade do serviço* (QoS). Tais atributos caracterizam serviços de entidades como aplicação, sistema operacional, rede e dispositivos de “hardware” que representam os blocos fundamentais da arquitetura de um sistema multimídia distribuído (SMD). Em ambientes melhor esforço (“best-effort”), alguns parâmetros de QoS, como a freqüência de quadros de vídeo e freqüência de amostras de áudio, podem sofrer variações grandes e descontroladas em seus valores durante a transmissão devido ao congestionamento de rede e a perdas de unidades de transporte. Tais variações afetam grandemente a satisfação do usuário com relação à QoS percebida. Assim, as aplicações multimídia distribuídas devem ser controladas por mecanismos de adaptação que ajustem a taxa de bits à carga da rede.

Geralmente, há dois tipos de adaptação de QoS [4]: 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 fornecida pela rede, que informa sua carga corrente através de um ou mais parâmetros da camada de comunicação como, por exemplo, a taxa de perdas de pacotes. Nesse caso, a informação de uma taxa de perdas baixa leva o mecanismo de adaptação a aumentar lentamente a taxa de bits gerada pela aplicação, enquanto uma taxa de perdas elevada conduz a uma diminuição desta.

Contudo, a maior parte das abordagens para regulação da taxa de bits não contempla um importante aspecto: *a considerável incerteza presente na determinação do estado da rede*. Tal incerteza é inerente à dinâmica dos SMD's, especialmente em WAN's - nas quais usuários entram e saem com grande freqüência, aplicações de tempos em tempos emitem rajadas (“burst”), conexões caem etc. - tornando o comportamento do sistema completamente imprevisível.

Busse, Deffner e Schulzrinne propõem em [3] um mecanismo de adaptação realimentado pela taxa de perdas de pacotes, obtida através dos pacotes RR (“receiver report”) fornecidos pelo Protocolo de Transporte de Tempo Real (“Real-Time Transport Protocol” - RTP). Três intervalos precisos de valores de perdas de pacotes definem o estado da rede e o ajuste da taxa de bits é feito através de funções lineares. Em [12], Sisalem descreve um mecanismo semelhante, também baseado no uso do RTP e de funções lineares, chamado Algoritmo de Ajuste Direto (“Direct Adjustment Algorithm” - DAA). Nesse mecanismo, contudo, a rede pode ser vista em apenas dois estados.

Em um trabalho anterior [6], nós propusemos um mecanismo de adaptação realimentado, com algumas características do mecanismo de Busse e do DAA, mas com os estados da rede sendo representados através de conjuntos nebulosos e o cálculo da taxa de bits feito por um controlador nebuloso do tipo Takagi-Sugeno [14]. Nesse trabalho, foi também discutida a adequação da abordagem proposta em virtude do aspecto da incerteza levantado acima. No presente trabalho, nós detalhamos alguns aspectos do mecanismo de adaptação nebuloso e descrevemos sua implementação. Os resultados obtidos em testes do controlador nebuloso foram comparados com aqueles obtidos através das implementações do mecanismo proposto por Busse et al. e do DAA, mostrando que

nosso mecanismo fornece um ajuste mais suave da taxa de bits, adequado para aplicações executadas sobre redes onde a carga varia tanto de forma brusca quanto gradual. O mecanismo proposto foi testado considerando-se aplicações de distribuição de vídeo, mas pode ser usado também para adaptação de áudio alterando-se os valores-limite dos parâmetros relacionados à taxa de perdas.

O restante deste trabalho é estruturado da seguinte forma: na próxima seção, é fornecida uma breve introdução ao protocolo RTP; na Seção 3, são discutidos alguns trabalhos existentes relacionados a mecanismos para ajuste de taxa de bits e ao uso de controle nebuloso para adaptação de QoS; a Seção 4 descreve o mecanismo de controle nebuloso proposto; a Seção 5 apresenta uma comparação de nossa abordagem com as duas outras propostas mencionadas acima; a Seção 6 resume nossos resultados e apresenta propostas para trabalhos futuros.

2 O Protocolo de Transporte de Tempo Real

O protocolo RTP é um protocolo fim a fim proposto pela “Internet Engineering Task Force” (IETF). Por não ter nenhuma noção de conexão, o RTP pode ser usado para operar tanto sobre protocolos de camadas inferiores orientados ou não à conexão. Ele não tem nenhuma dependência de formatos específicos de endereço e é executado tipicamente como parte da aplicação (e não como parte do núcleo do sistema operacional).

O RTP consiste de duas partes: uma parte de dados e uma parte de controle. Os pacotes de dados do RTP são formados por um cabeçalho de 12 bytes seguido pelo dado a ser transportado (“payload”), por exemplo, um quadro de vídeo ou uma seqüência de amostras de áudio. Dentre outras informações, o cabeçalho contém: tipo de “payload”, que identifica o tipo de informação carregada pelo pacote (por exemplo, vídeo comprimido no formato JPEG ou áudio comprimido no formato GSM); registro de tempo ou “timestamp”, que informa o instante da geração dos dados contidos no pacote; número de seqüência, que permite a detecção de perdas e a seqüencialização de uma série de pacotes com o mesmo “timestamp”; bit marcador, cuja interpretação depende do tipo do “payload” (para vídeo, esse bit marca a extremidade de um quadro; para áudio, ele marca o início de uma fala ou “talkspurt”); e identificador da fonte de sincronização (“synchronization source” - SSRC), um valor gerado aleatoriamente que identifica a fonte dentro de uma sessão.

A parte de controle é representada por um protocolo do controle chamado RTCP (“Real-Time Control Protocol”). 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 dos 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 dos receptores para os emissores, e SR (“sender report”), enviados dos emissores para os receptores. Diversos pacotes RR ou SR podem ser concatenados para formar um pacote composto RTCP, transmitido da camada de transporte na forma de um único pacote.

Os pacotes RTCP contêm muitas informações relacionadas à QoS. Uma vez que

eles são enviados na forma multiponto, cada membro da sessão pode saber como ela está transcorrendo para os demais participantes. As aplicações que têm emitido recentemente dados de áudio ou vídeo geram um pacote SR que contém informações úteis para a sincronização intermídia, bem como informações que permitem que os receptores estimem a taxa de bits real. Os membros da sessão emitem pacotes RR correspondentes a todos os dados de vídeo ou áudio recebidos recentemente. Esses pacotes contêm o número de seqüência mais elevado recebido, o número de pacotes perdidos, a variação do atraso e os “timestamps” necessários para estimar o atraso total (“round-trip delay”) entre o emissor e o receptor que enviou o pacote RR.

3 Trabalhos Relacionados

Antes de nosso trabalho, foram propostos dois outros mecanismos para ajuste dinâmico da taxa de bits a partir de informações fornecidas pelo RTCP. O primeiro, proposto por Busse et al. e descrito em [3], é um mecanismo onde a aplicação no lado do emissor usa os pacotes RR para calcular a taxa de perdas de pacotes. Baseado nesta métrica, o estado da rede, conforme ele é visto pelos receptores, é determinado e a taxa de bits é ajustada por um regulador linear com uma zona morta (“dead zone”). A taxa de 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 $i^{\text{ésimo}}$ -receptor ($perdas_i^f$) é calculada de acordo com a equação abaixo:

$$perdas_i^f(\tau_k) = (1 - \alpha) \times perdas_i(\tau_{k-1}) + \alpha \times perdas_i(\tau_k) \quad (1)$$

onde α é uma constante ($0 \leq \alpha \leq 1$), $perdas_i(\tau_{k-1})$ é a taxa de perdas do $i^{\text{ésimo}}$ receptor no tempo τ_{k-1} , e $perdas_i(\tau_k)$ é a taxa de perdas do $i^{\text{ésimo}}$ receptor no tempo τ_k . O incremento de α aumenta a influência de $perdas_i(\tau_k)$ enquanto seu decremento resulta em uma maior influência de $perdas_i(\tau_{k-1})$.

O valor de $perdas_i^f$ é usado pelo mecanismo para determinar o estado como a rede é vista pelo $i^{\text{ésimo}}$ -receptor: DESCARREGADA, CARREGADA ou CONGESTIONADA. Tais estados representam, de fato, três intervalos precisos: $[0, perdas_u[$, $[perdas_u, perdas_l[$ e $[perdas_l, 1]$. 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 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 Fig. 1 mostra o algoritmo proposto por Busse et al. onde N é o número total de receptores, N_c é o número de receptores no estado congestionado e N_l é o número de 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 (incluindo informações introduzidas pela pilha de protocolos e determinada a partir dos pacotes SR) 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. A Fig. 2 mostra o algoritmo simplificado, para o caso de somente um receptor.

```

if  $\left(\frac{N_e}{N}\right) \leq N_d$  then  $d \leftarrow DECREMENTA$ 
else if  $\frac{N_l}{N} \leq N_h$  then  $d \leftarrow MANTEM$ 
    else  $d \leftarrow INCREMENTA$ 
case  $d$ 
DECREASE:  $Bps' \leftarrow \max(Bps_r \times \nu, Bps_{min})$ 
INCREASE:  $Bps' \leftarrow \min(Bps_r + \eta, Bps_{max})$ 

```

Figura 1: Algoritmo de Busse para ajuste da taxa de bits

```

if  $perdas^f \geq perdas_l$  then  $Bps' \leftarrow \max(Bps_r \times \nu, Bps_{min})$ 
else if  $perdas_u \leq perdas^f < perdas_l$  then  $Bps' \leftarrow Bps_r$ 
    else if  $perdas^f < perdas_u$  then  $Bps' \leftarrow \min(Bps_r + \eta, Bps_{max})$ 

```

Figura 2: Algoritmo de Busse para apenas um receptor

O segundo mecanismo para ajuste de taxa de bits também baseado no uso do protocolo RTCP é o algoritmo DAA [12]. Nesse mecanismo, o 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}):

$$Bps' \leftarrow Bps_r + \frac{\eta}{\min(N, th_{scale})} \quad (2)$$

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:

$$th_{scale} = \frac{I_{min} \times Bps_{RTCP}}{S_{RTCP}} \quad (3)$$

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 receptor i informa uma taxa de perdas (filtrada como no mecanismo de Busse) maior do que o limite pré-definido ($perdas_c$), o emissor entra no estado congestionado e a taxa de bits é reduzida na seguinte proporção:

$$Bps' \leftarrow Bps_r \times (1 - perdas_i^f + perdas_c) \quad (4)$$

A identidade do 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 receptor j reporta valores de perdas maiores do que *MaisAltaPerda*, a taxa de bits sofre uma redução adicional de acordo com a equação abaixo:

$$Bps' \leftarrow Bps_r \times (1 - perdas_j^f + MaisAltaPerda) \quad (5)$$

MembroPerdedor recebe então a identidade do receptor j e *MaisAltaPerda* recebe $perdas_j^f$. Após receber uma mensagem de *MembroPerdedor* com valores de perdas menores do que $perdas_c$, o emissor volta para o estado descarregado. Como em [3], pode-se especificar as taxas de bits mínima e máxima permitidas. Para um receptor, o algoritmo pode ser simplificado de acordo com a Fig. 3.

```

if  $perdas^f > perdas_c$  then  $Bps' \leftarrow \max(Bps_r \times (1 - perdas^f + perdas_c), Bps_{min})$ 
else if  $perdas^f \leq perdas_c$  then  $Bps' \leftarrow \min(Bps_r + \eta, Bps_{max})$ 

```

Figura 3: Algoritmo DAA para apenas um receptor

3.1 Lógica Nebulosa em SMD's

Há poucos trabalhos que contemplam o uso de Lógica Nebulosa para adaptação de QoS. Em [2], 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 emissores via células RM (a ER define a taxa de bits máxima dos emissores). O controlador apresentado difere do nosso em vários aspectos: ele usa o modelo de Mamdani para controladores nebulosos, é amarrado à tecnologia ATM, sendo posicionado no “switch”, e usa a taxa de ocupação de “buffer” como entrada.

Em [10], Li e Nahrstedt propõem 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 em nosso trabalho em termos de variáveis e das regras usadas pelo controlador nebuloso, mas muitos aspectos teóricos apresentados nele podem ser usados como um auxílio para uma melhor formalização de nossa abordagem.

4 Mecanismo de Ajuste da Taxa de Bits Baseado em Controle Nebuloso

O mecanismo da adaptação de taxa de bits proposto neste trabalho regula a quantidade de dados injetado na rede segundo uma abordagem similar àquelas duas propostas apresentadas previamente. Contudo, em nossa proposta o estado da rede e a ação de controle são determinados de maneira imprecisa usando um controlador nebuloso. Nas sessões seguintes, serão apresentados alguns conceitos relacionados à Lógica Nebulosa bem como a estrutura geral de nosso mecanismo do controle.

4.1 Controladores Nebulosos

Um controlador nebuloso (CN) é um mecanismo de controle cujas origens encontram-se nas pesquisas de H. Mamdani [11]. Os CN's são baseados na Teoria dos Conjuntos

Nebulosos introduzida por Lotfi Zadeh em [17]. Formalmente, um conjunto nebuloso \tilde{A} do domínio (ou *universo de discurso*) X é definido pela função

$$\mu_{\tilde{A}} : X \mapsto [0, 1].$$

A função $\mu_{\tilde{A}}$ - chamada *função de pertinência* - associa um grau de compatibilidade com o conceito expresso por \tilde{A} a cada elemento $x \in X$: se $\mu_{\tilde{A}}(x) = 1$, x é completamente compatível com \tilde{A} ; se $\mu_{\tilde{A}}(x) = 0$, x é completamente incompatível com \tilde{A} ; e se $0 < \mu_{\tilde{A}}(x) < 1$, x é parcialmente compatível com \tilde{A} , com um grau $\mu_{\tilde{A}}(x)$.

Como qualquer função, a função de pertinência pode ser discreta ou contínua. No caso discreto, o universo de discurso X é representado através de um número de amostras aos quais são associados graus de pertinência arbitrários para os conjuntos nebulosos. Tais valores são intuitivos, baseados na experiência do projetista. No caso contínuo, os graus de pertinência são obtidos para todos os elementos de X usando-se alguma função matemática. Exemplos comuns de funções de pertinência contínua são: função triangular, função trapezoidal e função em forma de sino (“bell-shaped”). Tais funções requerem alguns parâmetros, também definidos de forma arbitrária.

Em nosso caso, o universo de discurso é o conjunto $\%perdas = [0, 1]$. Esse universo é dividido em 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 $x = 0.7$, $\mu_{DESCARREGADA}(x) = 0.00$, $\mu_{CARREGADA}(x) = 0.53$, $\mu_{CONGESTIONADA}(x) = 0.84$.

O CN usado por nosso mecanismo de ajuste está posicionado no lado do emissor e segue um esquema de acordo com o diagrama de blocos mostrado na Fig. 4. As seções seguintes explicam a função de cada um dos blocos.

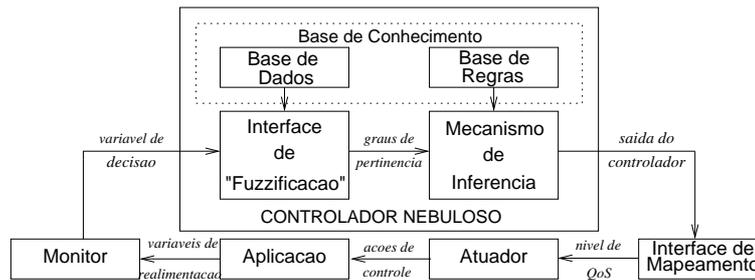


Figura 4: Diagrama de blocos do mecanismo de controle nebuloso

4.2 Monitor

O monitoramento consiste na obtenção da taxa de perdas de pacotes de cada receptor ($perdas_i \in \%perdas$, $i = 1, 2, \dots, N$) através dos pacotes RR mais recentes. Esse percentual é suavizado usando um filtro passa baixa como em [3] e [12]. As perdas suavizadas são usadas para calcular a taxa de perdas agregada (\overline{perdas}). Diferentes

políticas podem ser usadas para a obtenção de $\overline{\text{perdas}}$. Tais políticas influenciam fortemente a decisão de ajuste e, conseqüentemente, o impacto da ação de controle sobre a QoS recebida pelos receptores. Alguns exemplos de políticas são a média aritmética, a média harmônica, a média ponderada¹ e o pior caso.

4.3 Base de Regras e Mecanismo de Inferência

Basicamente, as regras tem o formato `if <condições> then <conclusão>`, sendo que se o sistema tem mais de uma entrada, as sentenças das condições são conectadas geralmente pelos operadores *e* e *ou*. As regras usadas pelo mecanismo são mostradas na Fig. 5 (o significado dos símbolos é o mesmo apresentado anteriormente na Seção 3).

```

if  $\overline{\text{perdas}}$  is CONGESTIONADA then  $Bps' \leftarrow \max(Bps_r \times \nu, Bps_{min})$ 
if  $\overline{\text{perdas}}$  is CARREGADA then  $Bps' \leftarrow Bps'$ 
if  $\overline{\text{perdas}}$  is DESCARREGADA then  $Bps' \leftarrow \min(Bps_r + \eta, Bps_{max})$ 

```

Figura 5: Base de regras para o CN

A base de regras acima segue o modelo de CN's de Takagi-Sugeno, no qual as saídas são combinações ou funções lineares das entradas (veja Fig. 5). De fato, cada regra representa um controlador linear e o CN faz a interpolação entre elas.

O mecanismo de inferência é o responsável pelo cálculo da saída do CN. Sua entrada é o valor de $\overline{\text{perdas}}$ convertido, pela interface de “fuzzificação”, para graus de pertinência através de uma pesquisa na base de dados que contém as funções da pertinência associadas aos três conjuntos nebulosos. A saída final (já na forma não-nebulosa) é o resultado da contribuição ponderada de cada regra, sendo dada pela equação:

$$\begin{aligned}
 Bps' = & \frac{\mu_{DESCARREGADA}(\overline{\text{losses}}) \times (Bps_r + \eta) + \mu_{CARREGADA}(\overline{\text{losses}}) \times Bps' +}{\mu_{DESCARREGADA}(\overline{\text{losses}}) + \mu_{CARREGADA}(\overline{\text{losses}}) +} \\
 & \frac{+ \mu_{CONGESTIONADA}(\overline{\text{losses}}) \times (\nu \times Bps_r)}{+ \mu_{CONGESTIONADA}(\overline{\text{losses}})} \quad (6)
 \end{aligned}$$

4.4 Interface de Mapeamento

Além do cálculo da taxa de bits adaptada ao estado da rede, o mecanismo de adaptação necessita mapear essa taxa para a aplicação 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 QoS. Uma vez que diferentes combinações de valores de parâmetros de QoS podem ter exigências similares de taxa de bits representando, contudo, qualidades completamente distintas, é necessário estabelecer um critério de escolha que permita ao mecanismo de adaptação a seleção da melhor combinação de valores destes parâmetros para uma dada taxa de bits. Neste trabalho, utilizamos a *função grau da qualidade* - introduzida por nós em

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

[8] - para fazer essa seleção. Tal função será descrita a seguir após a introdução de alguns conceitos.

Seja ρ_i um parâmetro de QoS genérico da camada de aplicação e $\Omega_{\rho_i} = [\rho_{i_{min}}, \rho_{i_{max}}]$ ($i = 1, 2, \dots, n$) o domínio de valores de ρ_i . Seja também $L = \langle \rho_1, \rho_2, \dots, \rho_n \rangle$ 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 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, nós definimos uma métrica de qualidade, chamada *grau de qualidade (QoS)*, arbitrariamente incluída no domínio $[0, 1]$. O valor de *QoS* é obtido usando-se a função grau de qualidade:

$$QoS : L \mapsto [0, 1]. \quad (7)$$

Para o nível de QoS $\overset{j}{L} = \langle \overset{j}{\rho}_1, \overset{j}{\rho}_2, \dots, \overset{j}{\rho}_n \rangle$, temos $QoS_{\overset{j}{L}} = QoS(\overset{j}{\rho}_1, \overset{j}{\rho}_2, \dots, \overset{j}{\rho}_n)$. A função grau de qualidade pode ser construída usando-se funções utilidade² para cada dimensão de *QoS*. Seja $v_{\rho_1}, v_{\rho_2}, \dots, v_{\rho_n}$ as funções utilidade e $\omega_{\rho_1}, \omega_{\rho_2}, \dots, \omega_{\rho_n}$ os pesos dos parâmetros de QoS da camada de aplicação $\rho_1, \rho_2, \dots, \rho_n$, onde esses pesos refletem a importância relativa de ρ_i de acordo com o tipo de aplicação³. Para o nível de QoS $\overset{j}{L} = \langle \overset{j}{\rho}_1, \overset{j}{\rho}_2, \dots, \overset{j}{\rho}_n \rangle$, temos:

$$QoS(\overset{j}{\rho}_1, \overset{j}{\rho}_2, \dots, \overset{j}{\rho}_n) = \min(v_{\rho_1}(\overset{j}{\rho}_1) \cdot \omega_{\rho_1}, v_{\rho_2}(\overset{j}{\rho}_2) \cdot \omega_{\rho_2}, \dots, v_{\rho_n}(\overset{j}{\rho}_n) \cdot \omega_{\rho_n}), \omega_{\rho_1} + \omega_{\rho_2} + \dots + \omega_{\rho_n} = 1 \quad (8)$$

As funções utilidade podem ser construídas usando-se os resultados de escores de opinião média (“mean opinion score” - MOS) como em [5]. A função *QoS* tem também uma dimensão que associa os requisitos em termos de consumo de recursos (largura de banda, no nosso caso) a cada nível de QoS.

A função *QoS* pode ser representada como uma tabela cujas entradas contêm níveis de QoS (como em [15]) mais o grau de qualidade (*QoS*) e a taxa de bits (*Bps*) associados. Seja, por exemplo, uma aplicação de transmissão de vídeo com apenas dois parâmetros de QoS da camada aplicação que podem ser dinamicamente configurados: frequência de quadros (*Fps*) e fator de quantização (q) com domínios $\Omega_{Fps} = \{0, 1, 2, \dots, 29, 30\}$ e $\Omega_q = \{1, 2, \dots, 29, 30, 31\}$ ($q = 1$ representa a melhor qualidade e $q = 31$ a pior). Neste caso, uma entrada da tabela é representado pela 4-tupla $\langle Bps, Fps, q, QoS \rangle$. *Bps* é obtido empiricamente e representa o pior caso para cada nível de QoS. A taxa de bits pode ser obtida através de medições sobre vídeos contendo muitos detalhes (o que diminui a taxa de compressão espacial); se o algoritmo de compressão realiza também compressão temporal (caso do MPEG-1, MPEG-2, MPEG-3, H261 e H263), os vídeos devem ser compostos apenas de quadros do tipo I. As entradas da tabela poderiam ser similares à $\langle 462.580000, 30, 1, 1.000000 \rangle$, $\langle 385.613079, 25, 1, 1.000000 \rangle$, $\langle 370.655038, 24, 1, 0.990000 \rangle$, ..., $\langle 111.996435, 5, 16, 0.000000 \rangle$, $\langle 110.621875, 5, 16, 0.000000 \rangle$, ..., $\langle 034.737054, 1, 31, 0.000000 \rangle$, $\langle 000.000000, 0, 31, 0.000000 \rangle$.

²Uma função utilidade [9] mapeia valores de um único parâmetro de QoS para uma utilidade subjetiva pertencente ao intervalo $[0,1]$, obtida a partir das preferências dos usuários.

³Em uma videoconferência comum, por exemplo, os pesos dos parâmetros de QoS relacionados ao som deveriam ser maiores do que aqueles relacionados à imagem.

Quando o controlador calcula uma nova taxa de bits Bps' , o mecanismo irá pesquisar na tabela o primeiro nível de QoS $L' = \langle Bps_{L'}, Fps_{L'}, q_{L'}, QoS_{L'} \rangle$ tal que $Bps_{L'} \leq Bps'$ (uma vez que a chave de ordenação da tabela é QoS , L' será o nível de QoS com o maior grau de qualidade que satisfaz a condição).

Pode-se observar que dois ou mais diferentes níveis de QoS podem ter o mesmo grau de qualidade. Tal comportamento da função é previsível: de fato, os usuários não podem perceber diferenças entre valores de QoS dentro de certos intervalos (a maioria de usuários, por exemplo, não percebe diferenças em frequências de quadros entre 25 e 30 fps). Assim, muitas vezes é possível para o mecanismo mudar a taxa de bits corrente sem mudar a qualidade, sob o ponto de vista do usuário final.

Outro aspecto importante a ser frisado é que o fato de um nível de QoS $\overset{i}{L}$ possuir um grau de qualidade maior do que $\overset{j}{L}$ não significa que $\overset{i}{L}$ requer mais largura de banda do que $\overset{j}{L}$. Este é o caso, por exemplo, dos níveis de QoS $\langle 370.655038, 24, 1, 0.990000 \rangle$ e $\langle 388.466882, 28, 10, 0.200000 \rangle$.

4.5 Atuador

O atuador é responsável pela configuração da aplicação, no lado do emissor, para que ela passe a emitir o fluxo com uma taxa de bits condizente com aquela calculada pelo CN. Para tal, o atuador configura os parâmetros de QoS da aplicação com os valores correspondentes ao nível de QoS L' . Para o exemplo anterior, o atuador mudaria os valores da frequência de quadros e fator de quantização da aplicação para $Fps_{L'}$ e $q_{L'}$ respectivamente. Tal mudança pode ser efetuada de duas maneiras: filtragem [16] e alteração dinâmica dos parâmetros do codificador [8]. A primeira abordagem pode ser usada tanto no caso de aplicações do tipo vídeo sob demanda, que usam dados armazenados (previamente codificados) quanto aplicações que usam dados “vivos” (como videoconferência), codificados durante a sessão; a segunda abordagem tem seu uso restrito às aplicações que usam dados vivos.

5 Discussão das Abordagens

As três abordagens apresentadas previamente fazem uso de intervalos ou faixas de valores de taxa de perdas quando da decisão de manter, incrementar ou decrementar a taxa de bits. Tais intervalos podem ser representados na forma de conjuntos, conforme a Fig. 6. Os intervalos usados pelo mecanismo de Busse podem ser vistos como conjuntos nebulosos que admitem apenas dois graus de pertinência⁴, 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 $trapmf(x, a, b, c, d)$, definida como:

$$trapmf(x, a, b, c, d) = \begin{cases} 0 & \text{se } x < a \\ \frac{x-a}{b-a} & \text{se } a \leq x < b \\ 1 & \text{se } b \leq x < c \\ \frac{d-x}{d-c} & \text{se } c \leq x < d \\ 0 & \text{se } x \leq d \end{cases} \quad (9)$$

⁴De fato, a Teoria dos Conjuntos “Clássicos” é um subconjunto da Teoria dos Conjuntos Nebulosos.

Visando facilitar a comparação das abordagens, os limites usados pelo mecanismo de Busse foram ajustados em $perdas_u = 0.02$ e $perdas_l = 0.04$ (valores apropriados para uma transmissão de vídeo segundo [3]) 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 $trapmf(\overline{perdas}, 0, 0, 0, perdas_u)$; para o conjunto CARREGADA, $trapmf(\overline{perdas}, 0, perdas_u, perdas_l, perdas_l + perdas_u)$; para o conjunto CONGESTIONADA, $trapmf(\overline{perdas}, perdas_l, perdas_l + perdas_u, 1, 1)$.

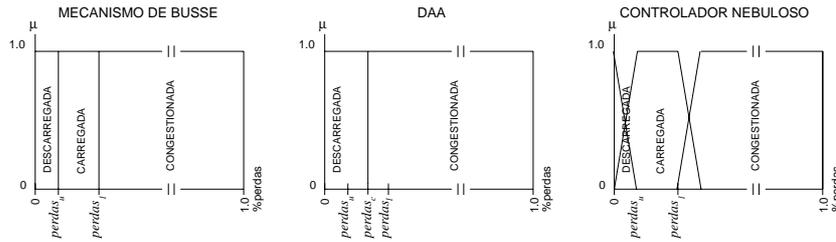


Figura 6: Funções de pertinência para os três mecanismos

Essa escolha garante que o mecanismo nebuloso terá a mesma zona morta do mecanismo de Busse. O uso de funções trapezoidais é justificado apenas pela simplicidade; funções de pertinência mais apropriadas para os conjuntos nebulosos podem ser obtidas usando-se, por exemplo, o método de agrupamento c-means [1].

Nas seções seguintes, os três mecanismos serão comparados em relação a três aspectos: comportamento da taxa de bits, justiça (“fairness”) no ajuste e escalabilidade.

5.1 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 QoS.

A Fig. 7 mostra os valores da taxa de bits calculada pelos três mecanismos em função de diferentes variações de perdas ($\Delta perdas = 0, 0.01, 0.02, \dots, 0.10$). 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 [3] e [12]). Na figura, pode-se verificar que o incremento da taxa de bits realizado pelo mecanismo de Busse e pelo 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 mecanismos de Busse 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 receptor.

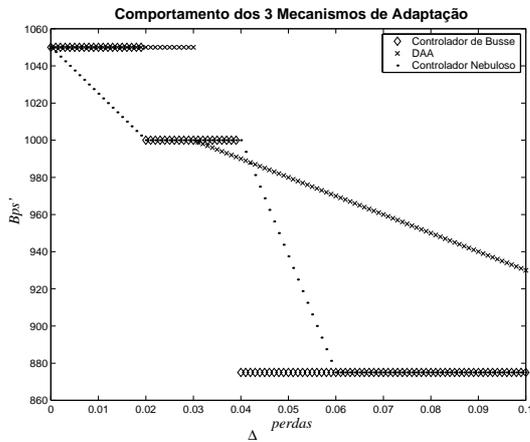


Figura 7: Taxa de bits calculada \times variação de perdas ($Bps_r = 1000$)

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 decréscimo 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.* 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 mecanismo de Busse e no 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.).

5.2 Justiça na Adaptação

Com relação à justiça, o problema ocorre principalmente quando os membros da sessão recebem dados através de ligações com capacidades muito diferentes, especialmente em conexões interurbanas. De acordo com [12], este problema - conhecido como “*beat down problem*” - ocorre em consequência do aumento da probabilidade de perdas a cada ligação adicional, fazendo com que receptores distantes reportem mensagens de controle com valores de perdas muito mais elevados do que aquelas fornecidas pelos receptores locais. Uma vez que a decisão do ajuste está fortemente relacionada às taxas de perdas relatadas, o mecanismo de controle deve usar uma política que tente

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 mecanismo de Busse 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 decréscimo da taxa de bits que desconsidera a taxa de perdas, pode conduzir a uma subutilização 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

ser o mais justa possível.

O mecanismo de Busse utiliza uma política 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 receptores estão descarregados. Assim como as políticas baseadas na média mostradas na Seção 4.2, essa política baseada na maioria dos casos garante que uns poucos receptores, conectados através de ligações com largura de banda baixa, não forçarão o emissor a fornecer uma qualidade baixa em detrimento dos demais 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 receptores mantenham-se congestionados por longo tempo, quando eles representarem a minoria. Essa política permite que, em um dado instante, até $N \times (1 - (N_d + N_h))$ receptores não tenham qualquer influência na decisão de adaptação.

O DAA utiliza uma política baseada no pior caso que garante que o mecanismo tentará manter todos os receptores descongestionados. A desvantagem dessa política é que um único 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 $N - 1$ receptores não tenham qualquer influência na decisão de adaptação.

Conforme visto, ambas políticas 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. Contudo, apesar de nosso mecanismo de adaptação não ser amarrado à nenhuma política específica para o cálculo de \overline{perdas} , acreditamos que, no caso genérico, uma escolha baseada na média é a política mais adequada, por ser a mais “democrática”, uma vez que ela possibilita que todos os receptores exerçam alguma influência na decisão de adaptação.

5.3 Escalabilidade

Com relação à escalabilidade, o principal problema é a largura de banda requerida pelo tráfego do controle em detrimento do tráfego de dados. Entretanto, quanto mais atualizada é a variável de estado, maior a largura de banda requerida pelo tráfego de controle. 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 mecanismo de Busse 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 Eq. 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 Fig. 5) para considerar o número de membros da sessão, utilizando, por exemplo, o fator th_{scale} como no DAA:

if \overline{perdas} is *DESCARREGADA* then $Bps' \leftarrow \min(Bps_r + \frac{\eta}{\min(N, th_{scale})}, Bps_{max})$

5.4 Avaliação Experimental dos Mecanismos

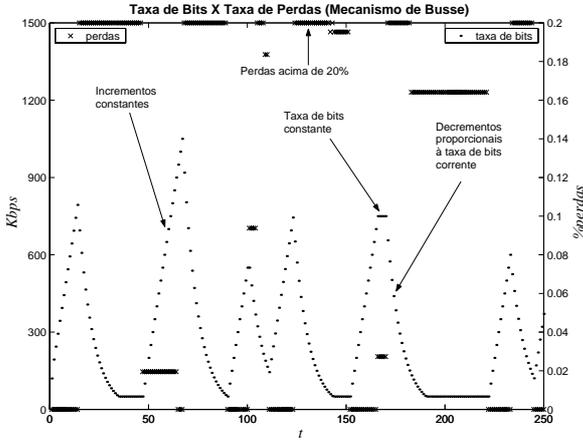


Figura 8: Experimentos sobre a Internet para o algoritmo de Busse

bits calculada pelo controlador. 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 receptor uma estação de trabalho na Universidade de Illinois, EUA. Os parâmetros usados nos três 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 JRTPLIB⁵, uma biblioteca orientada a objetos escrita em C++.

O monitor, posicionado no emissor, recebe os pacotes RTCP da aplicação no lado do receptor e envia ao controlador a taxa de perda de pacotes; a aplicação, no lado do emissor, envia um fluxo contínuo usando o RTP de acordo com a taxa de

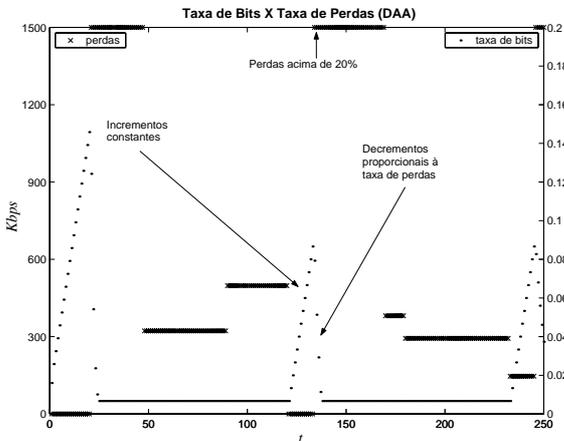


Figura 9: Experimentos sobre a Internet para o DAA

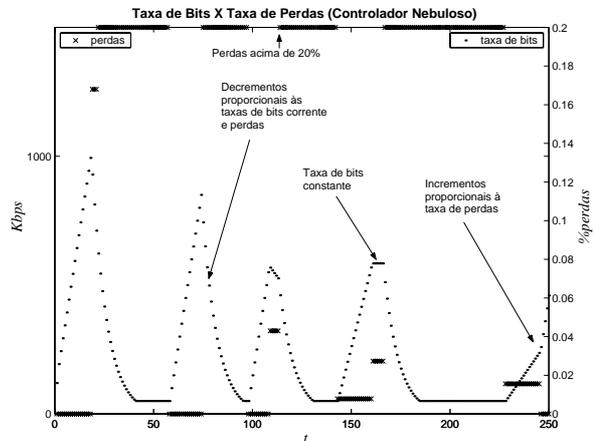


Figura 10: Experimentos sobre a Internet para o CN

foram configurados com os mesmos valores descritos no início da Seção 5. Para cada controlador, as medidas tiveram uma duração de 300 segundos. O parâmetro α do filtro passa baixa foi ajustado em 0.3 nos três casos (o valor mais adequado encontrado nos experimentos de Busse).

De acordo com os resultados obtidos e mostrados nas figuras 8 e 9, o incremento da taxa de bits⁶ no mecanismo de Busse (quando as perdas são menores que $perdas_u =$

⁵<http://lumumba.luc.ac.be/jori/jrtplib/jrtplib.html>

⁶A taxa de bits mostrada é aquela calculada pelo controlador e não a taxa de bits real.

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 Fig. 10, onde os pontos representando a taxa de transmissão calculada não são igualmente espaçados. Quando da degradação, as mudanças da taxa de bits do mecanismo de Busse 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 aquela proporcionada pelo mecanismo de Busse) 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 “distante” do ponto de vista do usuário final em virtude dos seguintes aspectos: (1) perdas baixas não indicam uma QoS 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 QoS 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; (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. Em [7], propomos, como critério de avaliação de mecanismos de controle de QoS, o grau de qualidade recebido, obtido através do uso da função *QoS* já apresentada, tendo como argumentos os valores dos parâmetros de QoS recebidos. Segundo nossa ótica, tal critério é mais adequado do que a taxa de perdas por refletir melhor o ponto de vista do usuário final.

6 Conclusão

Neste trabalho, nós apresentamos um mecanismo, posicionado no emissor, para ajuste dinâmico da taxa de bits de aplicações multimídia distribuídas. Tal ajuste é feito de acordo com o estado da rede, estimado de forma imprecisa por um CN a partir da taxa de perdas de pacotes informada pelos receptores. *A priori*, o uso de CN's para ajuste da taxa de bits parecia adequado devido à incerteza presente na determinação do estado

da rede. Os resultados obtidos em nossos experimentos comprovaram a adequação da abordagem proposta no sentido de proporcionar uma adaptação da QoS bem mais suave do que outras baseadas em controle não-nebuloso. A vantagem do uso da taxa de perdas de pacotes como variável de realimentação é que essa é uma das informações que pode ser obtida através do uso do RTCP, um protocolo de controle já bastante difundido e usado por várias aplicações multimídia. A desvantagem é que tal variável representa um parâmetro de QoS “distante” do usuário final.

Um problema desta abordagem e, de forma geral, de todos os esquemas de adaptação baseados em troca de informação entre emissores e receptores, é a possibilidade de ocorrência de uma explosão de realimentação, para um número demasiado grande de membros. Uma alternativa para diminuir essa possibilidade é através do uso de diversos controladores posicionados nos nós intermediários da rede, ao invés de um só no emissor. Essa abordagem reduz o tráfego de controle, mas aumenta o atraso, uma vez que os pacotes têm que ser montados nos nós para que seja realizada a adaptação.

Neste trabalho nós também descrevemos uma função que associa a taxa de utilização de recursos e as preferências do usuário - representadas através do conceito grau de qualidade - às combinações de valores de parâmetros de QoS da camada de aplicação ou níveis de QoS. Em nosso mecanismo de controle de QoS, essa função é usada para mapear a taxa de bits calculada pelo CN para um nível de QoS. O grau de qualidade pode também ser usado como variável de realimentação, como em [7], tornando o processo da adaptação ainda mais próximo da perspectiva do usuário final.

Devido ao fato do uso de controle nebuloso em SMD não ser ainda muito comum, há questões passíveis de serem objeto de investigações futuras, entre as quais: incorporação na base de regras de outras entradas (por exemplo, atraso, como em [13]), 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 construção da função grau de qualidade.

Referências

- [1] BEZDEK, J. C. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, 1981.
- [2] BOGATINOVSKI, M., TRAJKOVSKI, G., AND SPASENOVSKI, B. Fuzzy Controller for Video Conference Traffic in B-ISDN. In *IEEE 6th International Workshop on Computer-Aided Modeling, Analysis and Design of Communication Links and Networks (CAMAD'98)* (São Paulo, Brazil, 1998), pp. 55–59.
- [3] BUSSE, I., DEFFNER, B., AND SCHULZRINNE, H. Dynamic QoS of Multimedia Applications Based on RTP. In *1st International Workshop on High Speed Networks and Open Distributed Platforms* (St. Petesburg, Russia, May 1995).
- [4] GECSEI, J. Adaptation in Distributed Multimedia Systems. *IEEE Multimedia* (April-June 1997), 58–95.
- [5] GHINEA, G., AND THOMAS, J. QoS Impact on User Perception and Understanding of Multimedia Video Clips. In *6st ACM International Conference on Multimedia '98* (Bristol, England, September 1998), pp. 49–54.

- [6] KOLIVER, C., FARINES, J.-M., AND FRAGA, J. S. Controle Dinâmico de QoS Baseado no Uso do Protocolo RTCP e Lógica Difusa. In *VI Simpósio Brasileiro de Multimídia e Sistemas Hiperâmídia (SBMidia'2000)* (Natal, Brazil, June 2000).
- [7] KOLIVER, C., FARINES, J.-M., AND FRAGA, J. S. A RTP/Fuzzy Logic-Based Dynamic QoS Control. Tech. Rep. DAS2000-01, Federal University of Santa Catarina, Brazil, January 2001.
- [8] KOLIVER, C., FARINES, J.-M., FRAGA, J. S., AND REIS, H. L. Um Modelo para Adaptação de QoS Orientado ao Usuário Final. In *XVIII Simpósio Brasileiro de Redes de Computadores (SBR'2000)* (Belo Horizonte, Brazil, May 2000).
- [9] KRASIC, C., AND WALPOLE, J. QoS Scalability for Streamed Media Delivery. Tech. Rep. CSE-99-011, Oregon Graduate Institute of Science and Technology, Oregon, USA, September 1999.
- [10] LI, B., AND NAHRSTEDT, K. A Control-based Middleware Framework for Quality of Service Adaptation. *IEEE Journal on Selected Areas in Communications (JSAC)* 17, 9 (September 1999), 1632–1650.
- [11] MAMDANI, E. H., AND BAAKLINI, N. Prescriptive Method for Deriving Control Policy in a Fuzzy Logic Controller. *Electronic Letters* 11 (1975), 625–626.
- [12] SISALEM, D. Fairness of Adaptive Multimedia Applications. In *IEEE International Conference on Communications (ICC'98)* (Atlanta, USA, June 1998).
- [13] SISALEM, D., AND SCHULZRINNE, H. The Direct Adjustment Algorithm: a TCP-Friendly Adaptation Scheme. In *1st International Workshop Quality of Future Internet Services (QofIS'2000)* (Berlin, Germany, September 2000).
- [14] TAKAGI, T., AND SUGENO, M. Fuzzy identification of Systems and its Applications to Modeling and Control. *IEEE Transactions on Systems, Man and Cybernetics* 15, 1 (1985), 116–132.
- [15] VOGEL, A., KERHERVÉ, B., VON BOCHMANN, G., AND GECSEI, J. Distributed Multimedia Applications and Quality of Service: a Survey. *IEEE MultiMedia* 2, 2 (Summer 1995), 10–19.
- [16] YEADON, N., GARCIA, F., HUTCHINSON, D., AND MAUTHE, A. Filters QoS Support Mechanisms for Multipeer Communications. *IEEE Journal on Selected Areas in Communications (JSAC)* 14, 7 (September 1996), 1245–1262.
- [17] ZADEH, L. A. Fuzzy Sets. *Information & Control*, 8 (1965), 338–353.