

UMA CAMADA DE ADAPTAÇÃO À QUALIDADE DE SERVIÇO NA INTERNET PARA APLICAÇÕES MULTIMÍDIA

SEDIANE CARMEM LUNARDI
sediane@inf.pucrs.br

FERNANDO LUÍS DOTTI
fldotti@inf.pucrs.br

Pontifícia Universidade Católica do Rio Grande do Sul
Programa de Pós-Graduação em Ciência da Computação - PUCRS
Av. Ipiranga, 6681, Prédio 16
Porto Alegre – RS
90619-900, Brasil
<http://www.inf.pucrs.br>

RESUMO

A Internet tem sido utilizada, desde seu surgimento, para transmissão confiável de dados sem garantias de atraso, variação de atraso ou largura de banda. Contudo, o tráfego multimídia de aplicações que estão surgindo na Internet possui características e requisitos de QoS (Qualidade de Serviço) diferentes do tráfego convencional de dados discretos (p.e., texto e imagens). Várias abordagens têm sido consideradas para tentar garantir QoS. Entre essas, o desenvolvimento de aplicações com capacidade de modificar seu comportamento conforme o desempenho oferecido pela rede. Dentro deste contexto, este artigo apresenta o desenvolvimento de uma Camada de Adaptação de QoS para incorporar QoS em aplicações multimídia, visando melhorar a qualidade dessas aplicações no cenário de operação atual da Internet. A Camada possui funções de monitoração de QoS através de pacotes de controle RTCP (*Real Time Control Protocol*) e adaptação de QoS através da aplicação de políticas adaptativas. Como resultado, uma *Application Programming Interface* (API) foi desenvolvida. Esta API oferece funções para evitar a degradação das mídias que compõe a aplicação multimídia em cenários sem reserva de recursos como a Internet.

Palavras-chave: Aplicações multimídia, Qualidade de Serviço (QoS), Camada de Adaptação de QoS, Monitoração, Adaptação.

ABSTRACT

Since its beginning, the Internet has been used for reliable data transmission without delay, jitter or bandwidth guarantees. If compared to conventional discreet data traffic (e.g., text and images), the multimedia traffic of emerging applications over the Internet have different characteristics and QoS (Quality of Service) requirements. Several approaches have been developed trying to guarantee QoS. The development of applications that have the capability to adapt its behavior, depending on the network performance, is one example of such approaches. Within this context, this paper introduces the development of a QoS Adaptation Layer which incorporates QoS in multimedia applications with the purpose of improving their quality over the current Internet. This layer has QoS monitoring functions through RTCP packages and QoS adaptation functions through the application of adaptation policies. As a result, an Application Programming Interface (API) was developed. This API considers the scenario without resource reservation like the current Internet and offers functions to avoid the degradation of the various media *streams* composing a multimedia application.

Keywords: Multimedia applications, Quality of Service (QoS), QoS Adaptation Layer, Monitoring, Adaptation.

1 Introdução

O contínuo crescimento de aplicações multimídia na Internet tem despertado interesse não só de pesquisadores, como também de profissionais da área de redes de computadores e usuários em geral. Entusiasmo esse justificado pela promessa de aplicações de grande utilidade como educação a distância, rádio e TV sob demanda, videoconferências, entre outros. Aplicações como essas trazem novos requisitos de serviços normalmente expressos como

Qualidade de Serviço (QoS). O conceito de QoS é utilizado para especificar o conjunto de requisitos dessas aplicações através de parâmetros (p.e., atraso, variação do atraso, largura de banda e taxa de erros) [22]. Atender a esses requisitos é uma das principais razões de pesquisa na área de Qualidade de Serviço.

Ao longo dos últimos anos, pesquisas têm sido realizadas em torno de trabalhos relacionados à QoS. Na comunidade Internet esforços tem sido realizados para oferecer o modelo de Serviços Integrados (IntServ), o qual estende os serviços básicos IP com reserva de recursos por meio do protocolo RSVP — *Resource reSerVation Protocol* [3], e o modelo de Serviços Diferenciados (DiffServ), no qual os pacotes que trafegam por um domínio que implementa esta noção recebem tratamento diferenciado de acordo com sua necessidade [1]. Além disso, protocolos também têm sido desenvolvidos para assegurar serviços multimídia, em especial os protocolos *Real Time Transport Protocol* (RTP), para transporte de dados [16]; *Real Time Control Protocol* (RTCP), para controlar informações de *feedback* dos recebedores dos dados [16]. Em universidades trabalhos também vêm sendo desenvolvidos para suportar a noção de QoS com ou sem reserva de recursos.

Apesar desses esforços em oferecer garantias de QoS fim-a-fim, buscando melhorar a atual arquitetura Internet, seja estendendo seu modelo incorporando reserva de recursos, oferecendo serviços diferenciados, ou ainda, provendo novos protocolos e novas arquiteturas de comunicação, ainda existem vários fatores que contribuem para que aplicações multimídia não tenham seus requisitos de QoS atendidos de forma satisfatória, entre eles: heterogeneidade dos sistemas finais fazendo com que uma mesma aplicação se comporte de maneira diferente dependendo da máquina onde é executada; diferença de características (confiabilidade, atraso, vazão) do conjunto de redes que compõe a Internet influenciando à percepção do usuário final da aplicação multimídia; oferecimento de reserva de recurso "tudo ou nada", no qual, ou os recursos necessários para uma aplicação são oferecidos, ou nada é oferecido; probabilidade de reservas não serem suportadas em grandes partes da Internet ainda que por algum tempo; reserva de recursos ser cara em termos de troca de mensagens e possuir tarifação agregada.

Frente aos aspectos considerados, uma abordagem complementar é a implementação de aplicações capazes de ajustar seu funcionamento às condições variáveis que a rede de comunicação oferece, adaptando-se, assim, ao que lhe é oferecido. Vários trabalhos encontrados na literatura utilizam-se desta mesma abordagem, como em Bolot [2], Busse *et. al.* [4], Hutchison *et. al.* [8], Marakby *et. al.*[12], Sisalem [18], entre outros. Com isso, a responsabilidade em oferecer QoS para a aplicação multimídia passa a ser, também, do desenvolvedor da mesma. Isso é uma dificuldade para desenvolvedores de aplicações justamente pela falta de uma Interface para Programas de Aplicação — *Application Programming Interface* (API) — que enderece a abordagem de QoS fim-a-fim. APIs baseadas em QoS são necessárias para ocultar das aplicações a complexidade em administrar e controlar QoS.

Desta forma, o trabalho apresentado neste artigo visa tratar os novos requisitos das aplicações multimídia em cenário adaptativo, como a Internet. O objetivo principal é compensar alguns dos efeitos resultantes dos problemas encontrados na Internet relacionados ao suporte de QoS para aplicações multimídia através do desenvolvimento de uma Camada de Adaptação de QoS. A Camada foi desenvolvida e apresenta como resultado uma API que permite incorporar QoS em aplicações multimídia facilitando, dessa forma, a tarefa do desenvolvedor da aplicação na realização de tal atividade. A API proporciona interfaces, classes e métodos que permitirão ao desenvolvedor incorporar em uma aplicação multimídia funcionalidades de monitoração de QoS da rede, bem como adaptação da aplicação à qualidade recebida pela rede, fazendo com que esta se ajuste ao seu comportamento. A API pode ser usada em aplicações Java que utilizem

a API *Java Media Framework* (JMF) [10] e o protocolo RTP para transporte de seus dados (*Real Time Transport Protocol*) e RTCP (*Real Time Control Protocol*) para troca de informações de controle entre fonte e destino, sendo que a utilização desses protocolos é um requisito fundamental para a sua efetiva operação. Considerou-se em especial durante a fase de experimentos comunicação *unicast*, mas a API pode ser utilizada também para comunicação *multicast*.

Este artigo é organizado como segue. Na seção 2, trabalhos relacionados com aplicações adaptativas são descritos, haja vista que o trabalho aqui apresentado insere-se nesse contexto. Na seção 3, a camada de adaptação de QoS é apresentada e o cenário de validação segue na seção 4. Na seção 5, um mecanismo de adaptação é apresentado juntamente com os resultados alcançados, e na seção 6, algumas considerações finais são realizadas.

2 Trabalhos Relacionados

Aplicações adaptativas são chamadas “*network-aware*”, justamente pelo fato de adaptarem seu funcionamento às variações de desempenho do ambiente de rede [7]. Não requerem nenhum suporte especial da rede como por exemplo, controle de admissão e algum mecanismo para fornecer reserva de recursos, sendo apropriadas para utilização no ambiente Internet [9]. Existem para preencher uma lacuna entre o que a rede oferece e o que a aplicação espera dela, isto é, “reagir” às variações de desempenho da rede, mantendo um certo nível de qualidade para a aplicação. Desta forma, alguns trabalhos que utilizam esta abordagem são descritos abaixo.

Em Busse *et al.* [4] é apresentado um mecanismo para ajuste dinâmico dos requisitos de largura de banda para aplicações multimídia. A aplicação enviadora utiliza mensagens RTCP *Receiver Reports* (RR) para computar perda de pacotes, e baseado nessa métrica, o estado de congestionamento dos receptores é determinado. Conseqüentemente, um ajuste na largura de banda é realizado por meio de um algoritmo específico para este fim. O mecanismo é utilizado pela ferramenta de videoconferência Vic, e vários experimentos foram realizados na Internet e em uma rede ATM local.

Marakby *et al.* [12] descrevem uma arquitetura que torna possível transferência de mídias contínuas sobre a Internet utilizando o protocolo RTP. Na abordagem utilizada, existe um controlador de taxa adaptativo que ajuda a reduzir o congestionamento da rede através de informações de *feedback* do protocolo RTCP, ou seja, a taxa de transmissão dos dados é modificada para que a aplicação se ajuste ao estado corrente da rede. Uma aplicação multimídia adaptativa foi implementada para transferência de áudio e vídeo digital utilizando a abordagem proposta.

Outros trabalhos também relacionados ao ajuste dinâmico da taxa de envio da aplicação de acordo com o nível de congestionamento observado na rede podem ser encontrados em Sisalem [18] e Hutchison *et al.* [8]. No primeiro, o autor discute como os emissores podem aumentar sua taxa de envio durante situações em que a rede está levemente carregada, e como reduzi-la durante períodos de carga. Apresenta-se um algoritmo de ajuste que responde as flutuações da rede relacionada à largura de banda disponível mantendo baixa a taxa de perdas. Enquanto que no segundo, uma arquitetura e a implementação de uma aplicação multimídia adaptativa, que inclui o uso do protocolo RTP e seu protocolo de controle associado (RTCP), é discutida. A aplicação faz uso das informações de *feedback* RTCP para monitoração de QoS do tráfego na rede. As informações de *feedback* são utilizadas por um controlador de taxa adaptativo que é disparado no emissor pelo parâmetro de QoS fração de perda de pacotes. O

controlador tem por função ajustar e controlar a taxa de transmissão dos dados para aliviar o problema de congestionamento reagindo dinamicamente às modificações nas condições da rede.

Um "terminal de QoS" para aplicações adaptativas é discutido em Moghe *et al.* [14]. Os autores esperam que a próxima geração de terminais (p.e., redes de computadores, computadores pessoais) suporte sofisticadas aplicações adaptativas, pois alguns terminais possuem poder limitado e o atraso de processamento da aplicação pode ser um componente significativo do atraso fim-a-fim. Enfatizam que, atualmente, cada aplicação possui um algoritmo de adaptação nativo que opera independente de outras aplicações e de seus algoritmos de adaptação.

Em Bolot [2] é apresentado um mecanismo de *feedback* que adapta a taxa de saída de codificadores de vídeo (i.e., fontes de tráfego de vídeo) baseado no estado da rede. Esse mecanismo foi implementado em um codificador de vídeo H.261 para um sistema de videoconferência na Internet desenvolvido pelo INRIA (*French Institute for Research in Computer Science and Control*). Ele torna possível estabelecer e manter a qualidade da videoconferência quando congestionamentos ocorrem na rede, prevenindo também, que fontes de vídeo inundem os recursos da rede (p.e., *buffers* e links de saída). Vega-García *et al.* [21] utilizam um mecanismo de controle semelhante que adapta o processo de codificação e decodificação baseado nas características da rede. O objetivo é maximizar a qualidade do áudio transferida aos destinos; e, os autores descrevem e analisam um conjunto de mecanismos de controle. Eles incluem um mecanismo de controle para variação de atraso e um mecanismo de controle para taxa de erro que são combinados entre si. Os mecanismos foram implementados e avaliados sobre a Internet e o Mbone. Experiências realizadas demonstraram que é possível manter e estabelecer audioconferências com qualidade razoável frente a eventuais congestionamentos na rede.

3 A Camada de Adaptação de QoS

Como pôde-se perceber ao final da seção 2, várias aplicações vêm utilizando noções de adaptação como um meio de oferecer um nível mínimo de QoS às várias aplicações multimídia. Entretanto, nenhum suporte genérico é disponibilizado aos desenvolvedores desta classe de aplicações buscando facilitar seu trabalho em oferecer QoS para suas aplicações.

Implementar aplicações que adaptam-se ao comportamento variável da rede de comunicação é uma tarefa complexa para a maioria dos desenvolvedores de aplicações, implicando assim no tratamento de aspectos de comunicação como atraso, perda de pacotes, variação de atraso e vazão. Detalhes como obtenção e tratamento dos dados obtidos da rede de forma continuada, bem como modificações no comportamento da aplicação ocultando possíveis variações no desempenho de rede são algumas tarefas que fazem parte desse processo e que devem ser consideradas na implementação de tais aplicações. Isso não é uma tarefa trivial de ser realizada e requer habilidade do programador para tratar de questões desse tipo. Dessa forma, um suporte genérico para a construção de tais aplicações faz-se necessário; e, de maneira a preencher essa lacuna o trabalho aqui apresentado oferece uma API para auxiliar o desenvolvedor de uma aplicação multimídia qualquer na realização de tal atividade.

3.1 Arquitetura da Camada

Uma visão detalhada da arquitetura da Camada de Adaptação de QoS é ilustrada na Figura 1, a qual é constituída por um módulo de adaptação e outro de monitoração de Qualidade

de Serviço. A Camada, representada como uma camada funcional, trabalha sobre o protocolo para transporte de dados com propriedades de tempo real, RTP. Do ponto de vista de implementação, RTP pode ser considerado como parte integrante da aplicação pois executa em modo usuário acessando os recursos do sistema (interface *sockets*). Funcionalmente o protocolo RTP pode ser considerado como parte da camada de transporte, já que ele oferece serviços (p.e., *timestamps*, *sequence number*, *payload type*) às aplicações multimídia. No trabalho aqui apresentado, o protocolo foi considerado sob o último ponto de vista. Funcionalidades como monitoração da Qualidade de Serviço referente ao tráfego dos dados e adaptação da aplicação a qualidade recebida da rede são os serviços disponibilizados pela Camada de Adaptação de QoS à Camada de Aplicação da Arquitetura Internet.

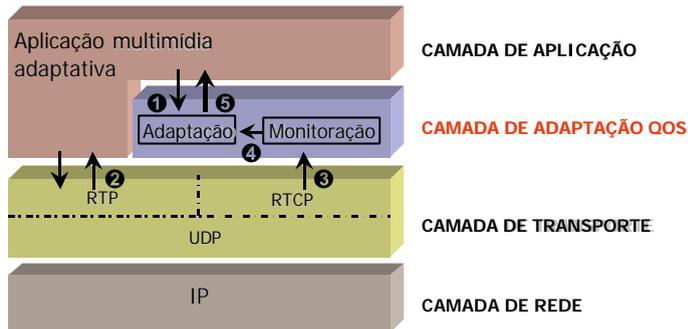


Figura 1 – Camada de Adaptação de QoS

As setas da figura acima (Figura 1) representam: (1) especificação de valores para os parâmetros de QoS da aplicação; (2) fluxo de pacotes de dados RTP (transmissão e recepção); (3) fluxo de pacotes de controle RCTP (*Sender Reports* e/ou *Receiver Reports*); (4) notificação sobre estado da rede, mais parâmetros de QoS (nível de rede); (5) gatilho para execução da(s) política(s) adaptativa(s).

O mecanismo geral de controle utilizado pela camada resume-se em análise de pacotes RTCP; estimativa do estado da rede; aplicação de políticas adaptativas em conformidade com o estado corrente da rede e em métricas realizadas com base nos parâmetros de Qualidade de Serviço especificados pela aplicação e obtidos da rede. Um aspecto importante é como o mecanismo de adaptação escolhido reage às decisões de estimativa do estado da rede. Essa reação depende das características da aplicação e do próprio controle interno da camada. Todos os passos, exceto a aplicação de políticas adaptativas, são independentes das características de QoS especificadas.

3.1.1 Monitoração de QoS

Monitoração de QoS é o processo de observar eventos relacionados com os requisitos de QoS da aplicação [23], reportando parâmetros de desempenho da rede e tornando possível identificar seu estado corrente. Logo, o processo de monitoração possui a responsabilidade de observar ações relacionadas com parâmetros de QoS e enviar mensagens para mecanismos que possam tratá-los. Dessa forma, esse indicará a ocorrência de violações quanto a QoS que está sendo oferecida pela rede.

Normalmente, o processo de monitoração está associado aos protocolos de transporte e é realizado por quem está recebendo os dados. Assim, o receptor pode monitorar a QoS e enviar *feedback* ao emissor sobre a recepção dos dados, informando número de pacotes perdidos, atrasos e suas variações. Na Camada de Adaptação de QoS, os parâmetros de QoS considerados

são a nível de rede, mais especificamente, parâmetros relacionados com o tráfego dos dados, ou seja, variação de atraso, atraso e perda de pacotes.

3.1.2 Adaptação de QoS

Adaptação de QoS é o processo no qual a aplicação torna-se capaz de ajustar-se às mudanças constantes das condições da rede [2]. Isso significa que a aplicação deve tolerar flutuações de operação da rede ajustando seu funcionamento ao que lhe é oferecido. Frequentemente, o processo de adaptação acontece como resultado de notificações emitidas por mecanismos de monitoração de Qualidade de Serviço. Estas notificações indicam mudanças no serviço que está sendo oferecido pela rede. Em aplicações adaptativas não existem protocolo de reserva de recursos, nem contrato de QoS com a rede.

Normalmente, mecanismos de adaptação de QoS possuem um conjunto de políticas adaptativas. Uma política adaptativa pode ser entendida como um conjunto de estratégias que tentarão assegurar à aplicação que essa terá pelo menos um nível mínimo de QoS. A regra de adaptação para QoS consiste em realizar ações corretivas baseadas na QoS da rede verificada e nos requisitos de QoS especificados pela aplicação. Mecanismos de adaptação de QoS são, normalmente, localizados no lado emissor de uma conexão; porém não está excluída a possibilidade do mecanismo localizar-se no lado receptor (ou qualquer um) [9]. Esses, podem ser associados ao transporte dos dados, a aplicação ou ao sistema de sinalização. Controle de fluxo (no transporte), adaptação da taxa de codificação de um vídeo MPEG-II (na aplicação) e renegociação de QoS (com sinalização) são exemplos de mecanismos. Para cada tipo de serviço — transporte, aplicação, sinalização — um mecanismo de adaptação (ou mais) é proposto baseado no tipo de violação de QoS [9]. Na Camada de Adaptação de QoS, os mecanismos de adaptação estão relacionados ao transporte dos dados e parcialmente a aplicação.

3.2 Funcionamento da Camada

Nesta subsecção será explicado o funcionamento dos módulos de monitoração e adaptação constituintes da Camada de Adaptação de QoS.

3.2.1 Módulo de Monitoração

O processo de monitoração, que tem por função supervisionar a qualidade da rede, é realizado na medida em que pacotes de controle, RTCP *Sender Reports* (SR) ou *Receiver Reports* (RR), vão sendo recebidos pela aplicação¹ e ao mesmo tempo detectados pelo módulo de monitoração. Desta forma, quando um pacote desse tipo chega (fonte ou destino dos dados), ele é detectado e tratado pelo módulo. Este tratamento, realizado a cada intervalo de recepção de um pacote de controle RTCP, consiste em:

1. Analisar cada *report block*, contido no pacote recebido, obtendo informações referentes aos indicadores de QoS à nível de rede associados ao tráfego dos dados, ou seja, parâmetros referentes a perda de pacotes, variação de atraso e atraso;

¹ O intervalo de chegada dos pacotes de controle pode variar de 5 segundos a 1 minuto (ou mais), dependendo do número de participantes, da largura de banda alocada para o tráfego RTCP e também do tamanho do pacote [19].

2. Calcular a taxa de perda de pacotes através do parâmetro de QoS fração de pacotes perdidos. A taxa de perdas é acumulativa desde o instante inicial do funcionamento do módulo, ou seja, início de transmissão ou recepção dos dados sobre a rede;
3. Verificar o estado atual da rede de comunicação com base no valor taxa de perdas calculado anteriormente. O estado da rede passa a ser então, ou rede carregada, ou rede não carregada, ou rede congestionada.

No receptor, apenas o seu *report block* é analisado; isso justifica-se porque o cliente está interessado somente nas informações que ele enviou ao receptor de seu *report*. No entanto, no lado emissor, todos os *report blocks* são analisados e, em especial, o parâmetro fração de pacotes perdidos de cada fonte RTCP RR é acumulado para verificar como todos os receptores estão sentindo a qualidade da rede.

Finalmente, com os parâmetros de QoS a nível de rede obtidos e estimado seu estado, uma notificação contendo essas informações é enviada ao módulo de adaptação de QoS. Esse último verifica se existe a necessidade de adaptação, e dependendo do caso, aplica a(s) política(s) escolhida(s) pelo desenvolvedor da aplicação. Cabe ressaltar que a monitoração do tráfego dos dados pode ser realizada tanto na aplicação enviada, quanto na aplicação receptora dos dados, sendo esta aplicada a cada mídia particular enviada ou recebida. O diagrama de estados representado pela Figura 2 ilustra o comportamento do módulo durante o funcionamento do sistema.

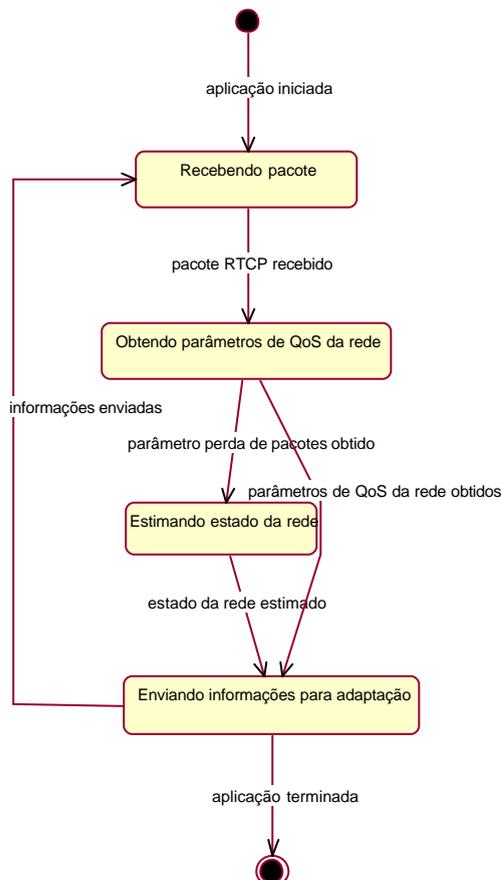


Figura 2 - Diagrama de eventos do módulo de Monitoração

3.2.1.1 Algoritmo para computar o estado da rede

O algoritmo utilizado neste trabalho é baseado na informação fração de pacotes perdidos encontrada em *receiver blocks* dos pacotes de controle RTCP.

a) Análise das mensagens RTCP

Somente a taxa de perda de pacotes é utilizada para identificar congestionamento na rede. Em Busse *et al.* [4], uma fórmula para calcular a taxa de perda, suavizar estatísticas e evitar oscilações de QoS é apresentada, e foi adotada para utilização neste trabalho. Essa fórmula é descrita pela Equação 1.

$$I^{atual} = (1 - a)I^{anterior} + ab$$

Equação 1 - Fórmula para cálculo da taxa de perdas

Onde,

b = novo valor referente a fração de perda de pacotes

a = fator de suavaização que determina o peso que é dado a taxa de perdas anterior

$0 \leq a \leq 1$

A rede é avaliada constantemente pelo algoritmo que verifica seu comportamento de acordo com a fração de pacotes perdidos, obtida através do *sequence number* contido no cabeçalho dos pacotes de dados RTP (no receptor). O estado anterior da rede sempre é levado em consideração a cada novo cálculo da taxa de perdas de modo a evitar tomadas de decisões de adaptação não necessárias em períodos curtos de pico da rede. Também, procurou-se descobrir um valor adequado para α . Segundo testes, o valor ótimo que resultou em uma taxa de perdas próxima da realidade foi de 0,5.

b) Estimativa do estado da rede baseado em perdas

O estado da rede é determinado como **carregado**, **não carregado**, **congestionado**. Dois limites são utilizados para isso², ou seja, até 2%, rede não carregada; de 2% à 4%, rede carregada; acima de 4%, rede congestionada. O processo funciona da seguinte forma, o valor perda de pacotes passa pelo filtro (fórmula dada pela Equação 1) e este computa a taxa de perda de pacotes. Depois deste cálculo, o estado da rede é determinado. O processo pode ser visualizado na Figura 3.

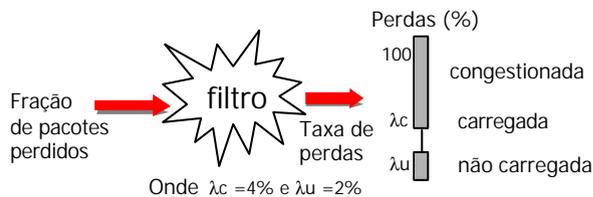


Figura 3 - Processo de obtenção do estado da rede

² Os estados e os valores limite são os mesmos utilizados em Busse *et al.* [4].

3.2.2 Módulo de Adaptação

O processo de adaptação, que tem por função adaptar a aplicação ao comportamento variável da rede, recebe uma especificação de QoS fornecida pela aplicação para cada mídia particular (valores limites para atraso, variação de atraso e perda de pacotes), juntamente com a política adaptativa que o sistema pode disparar para execução. É claro que um limite tolerável para que isso aconteça é dado à cada política particular pertencente ao sistema; e, após esse limite ser ultrapassado, a política é disparada e sua execução iniciada. Várias políticas podem ser escolhidas, cada qual com valores para os parâmetros de Qualidade de Serviço diferentes. Valores aproximados para diferentes tipos de mídias são encontrados em Schulzrinne [15], [17] e Campbell [5], os quais serviram como referência de utilização neste trabalho.

O tipo de adaptação utilizado depende da rede, ou seja, depois do módulo de monitoração receber *feedback* da rede, o módulo de adaptação aguarda por uma notificação, através do módulo de monitoração, que inclui o estado corrente da rede e os valores referentes aos parâmetros de qualidade oferecidos pela rede. Como consequência, os valores de QoS especificados pela aplicação são comparados com os valores de QoS da rede e, juntamente com seu estado, uma decisão acerca da(s) política(s) escolhida(s) é tomada resultando no disparo de sua execução ou não, conforme resultado de análise. Esta análise leva em consideração o parâmetro de QoS mais relevante para a política (p.e., perda de pacotes é o parâmetro avaliado quando a taxa de transmissão pode ser modificada). O diagrama de estados da Figura 4 descreve de forma mais detalhada o comportamento deste módulo no sistema.

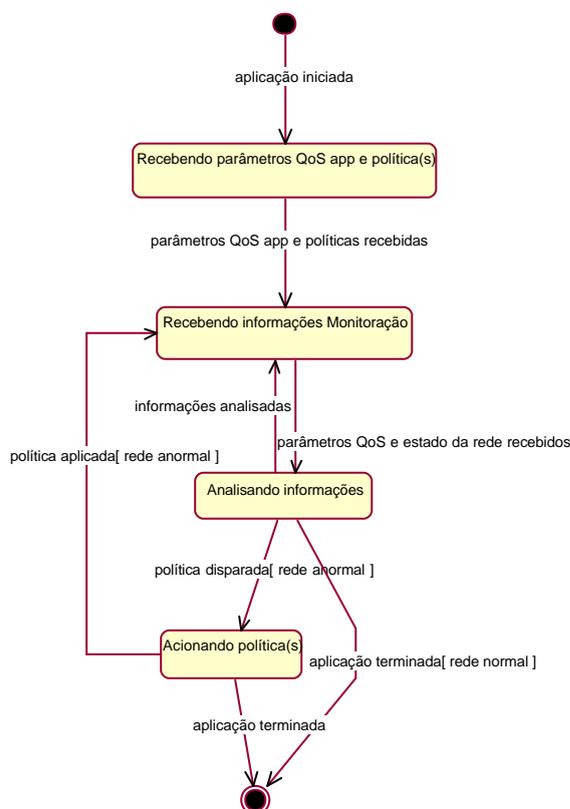


Figura 4 - Diagrama de estados do módulo de adaptação

Vários caminhos de adaptabilidade podem ser seguidos, como por exemplo, modificar a taxa de transmissão, modificar a codificação do áudio ou vídeo, modificar o tamanho do *frame*, modificar de colorido para preto e branco no caso de transmissão de vídeo, entre outros [6] apud [8]. As políticas escolhidas para implementação, as quais podem ser aplicadas tanto para áudio quanto para vídeo, foram: 1) modificação do tamanho do buffer no receptor visando compensar a variação de atraso entre os pacotes recebidos; 2) modificação da taxa de transmissão procurando reduzir o congestionamento da rede e melhorar a qualidade da mídia no receptor. Devido a sua importância, a última política adaptativa é descrita em mais detalhes na seção 5 deste artigo³.

Novas políticas de adaptação também poderão ser criadas pelo próprio desenvolvedor da aplicação de acordo com sua necessidade. Para tal, um modelo (classe) para construção de políticas é oferecido, além do suporte em tempo de execução (i.e., monitoração da QoS oferecida pela rede e adaptação da aplicação ao comportamento variável da rede). Quanto aos aspectos de sincronização entre diferentes mídias considerou-se que o protocolo RTP oferece as informações apropriadas para essa tarefa.

4 Cenário de Validação

A aplicação de validação escolhida para utilizar o conjunto de interfaces, classes e métodos disponibilizados pela API foi uma aplicação RTP, mais especificamente, uma sessão multimídia. A sessão multimídia de testes constitui-se de um Cliente RTP, que recebe dados através da rede, e um Servidor RTP, que transmite dados através da rede. Ambos utilizam o conceito de sessão RTP para isso.

Uma sessão RTP é uma associação entre um conjunto de participantes (aplicações) comunicando-se com RTP [16]. Um participante em uma sessão pode transmitir *streams*, receber *streams*, ou ambos. Para cada participante, a sessão é definida por um par particular de endereços de transporte destino (um endereço de rede mais um par de portas RTP e RTCP). O par de endereços de transporte destino podem ser comuns para todos os participantes, como é o caso de IP *multicast*, ou podem ser diferentes para cada participante, no caso de endereços de rede *unicast* individuais, mais um par de portas comuns. Em uma sessão multimídia, cada mídia é transportada em um sessão RTP separada com seus pacotes RTCP próprios [17]. Múltiplas sessões RTP são distinguidas pelos diferentes pares de números de portas e/ou diferentes endereços *multicast* [16].

4.1 Servidor RTP

Aplicações servidoras RTP transmitem *streams* de áudio e vídeo capturados ou armazenados através da rede (Figura 5). *Streams* podem ser codificados em múltiplos formatos de mídia e enviados em várias sessões RTP com receptores heterogêneos (p.e., uma sessão de conferência) [16]. Entretanto, antes dos *streams* RTP serem enviados sobre a rede, o formato do dado, durante a fase de processamento, deve ser modificado para um dos formatos de codificação específicos RTP apropriados para transmissão⁴. Assim, para cada *stream* de áudio ou vídeo a ser transmitido uma sessão RTP é criada.

³ Para saber detalhes sobre a política que modifica o tamanho do buffer no receptor veja LUNARDI [11]

⁴ A codificação normal recebe outro nome quando é utilizado dentro de RTP [17].

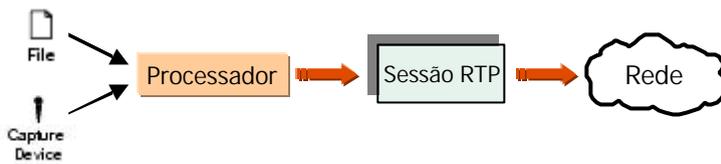


Figura 5 – Cenário de transmissão de um único *stream* RTP. *Stream* RTP é definido como uma série de pacotes que originam-se de uma fonte particular

4.2 Cliente RTP

Aplicações clientes RTP recebem *stream* através da rede e para cada novo *stream* recebido uma sessão RTP é criada. Os *streams* RTP recebidos são colocados em um *buffer* para posterior consumo pelo Player, o qual tem por função processar o *stream* de entrada e apresentá-lo em dispositivos de saída (monitores e alto-falantes). O processo de recepção de um fluxo de *streams* RTP pode ser visualizado na Figura 6.

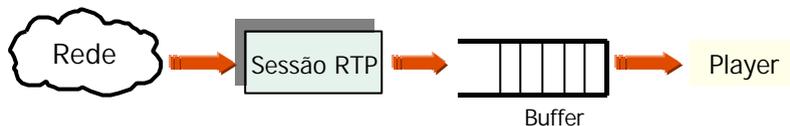


Figura 6 - Recepção de um fluxo de dados RTP

A aplicação servidor RTP transmite ambos *streams* de áudio e vídeo ao mesmo tempo e a aplicação cliente RTP recebe ambos. As funcionalidades referentes à monitoração de tráfego e adaptação de QoS foram acopladas, em cada sessão RTP criada, aos *streams* de áudio e vídeo respectivamente.

4.3 Experimentação

Como parte do cenário de validação e para realizar os experimentos necessários foram construídos dois monitores de tráfego contendo estatísticas globais de transmissão e recepção dos dados. Os dois monitores foram implementados como *threads*, as quais em execução têm por função coletar informações da rede em períodos de um segundo, permanecendo ativas durante todo o tempo de vida da sessão RTP na qual forem utilizadas. Eles foram utilizados para verificar se realmente aconteciam modificações no tráfego da rede quando da execução das políticas de adaptação, especificamente da política responsável por causar modificações no tráfego dos dados.

Também, para causar modificações de tráfego construiu-se um processo responsável por provocar perda de pacotes, já que optou-se por realizar testes em uma rede local. O processo foi implementado como uma *thread* de recepção e envio de pacotes UDP localizando-se assim entre o servidor RTP e o cliente RTP (Figura 7). Os experimentos foram realizados nos Laboratórios de Mestrado e Processamento Paralelo e Distribuído (LPPD) da PUCRS em redes Ethernet locais de 10MBps.

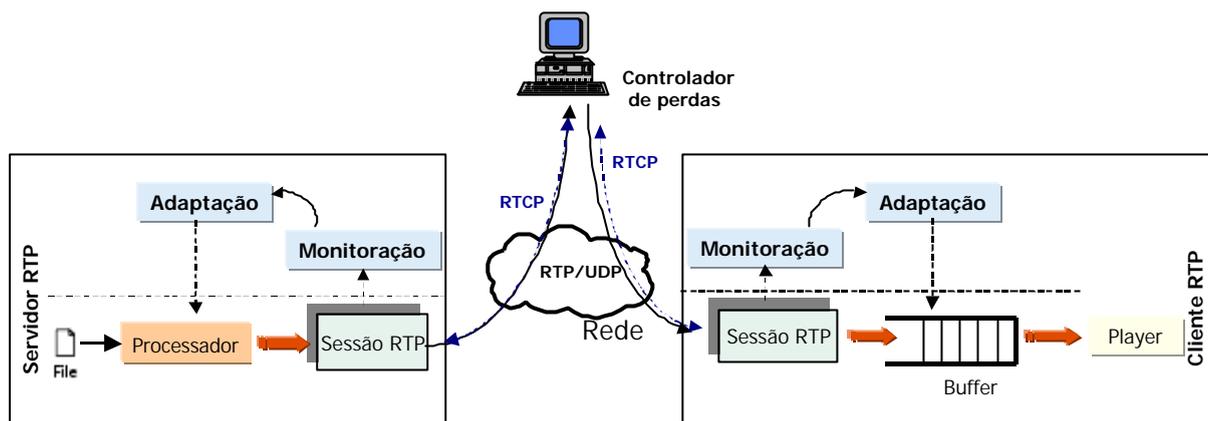


Figura 7 - Processo responsável por causar perda de pacotes RTP

5 Políticas de Adaptação de QoS

Políticas de adaptação são uma forma de modificar o comportamento da aplicação para que esta funcione da melhor maneira possível com os parâmetros de desempenho que a rede pode oferecer, tentando-se, desta forma, assegurar à aplicação que esta terá pelo menos um nível mínimo de QoS. Isso em conformidade com os valores dos parâmetros de qualidade especificados pelo desenvolvedor da aplicação. Estes valores servem como um limitador para que algumas medidas corretivas sejam realizadas. Abaixo a política que causa modificações no tráfego da rede é apresentada.

5.1 Modificar a taxa de transmissão dos dados

Perdas excessivas de pacotes, resultado de congestionamento da rede, podem causar significativa degradação da qualidade percebida do vídeo decodificado no receptor [13], bem como da qualidade de um áudio podendo torná-lo incompreensível. Em casos nos quais perdas e atrasos são causados por congestionamento da rede, uma diminuição na taxa dos dados transmitidos pode solucionar o problema de congestionamento da rede [20]. Além disso, pode ajudar a melhorar a qualidade do sinal decodificado no receptor, pois menos dados são perdidos durante a transmissão. Um caminho que pode ser utilizado para modificar o tráfego da rede é permitir que a fonte dos dados receba *feedback* sobre a qualidade da recepção dos dados, estime o estado corrente da rede (item 3.3.1) e controle a taxa na qual os pacotes são enviados através da rede. Em resumo, causar modificações na carga da rede com base nas condições em que ela se encontrar.

5.1.1 Implementação

A abordagem utilizada para modificar a taxa de transmissão durante uma sessão RTP foi alterar o formato do *stream* durante sua transferência sobre a rede utilizando-se da mesma sessão RTP para tal. Essa abordagem foi adotada com o intuito de ajudar a minimizar a largura de banda utilizada pela aplicação. Assim sendo, uma vez que os parâmetros de QoS da rede esperados não são atendidos, a política de adaptação troca o formato do *stream* que está sendo transmitido por outro de modo a economizar recursos da rede.

O processo de permutação entre formatos pode acontecer várias vezes durante a transmissão dos dados para o destino, dependendo das condições em que a rede se encontrar e

do número de formatos disponíveis de vazão inferior ao *stream* que está sendo enviado. A política pode ser aplicada várias vezes desde que a vazão do novo *stream* seja menor do que o atual. Cabe salientar que o formato do *stream* é modificado gradativamente durante a transmissão completa do arquivo de áudio ou vídeo.

5.1.2 Resultados de experimentos

O gráfico da Figura 8 ilustra a transmissão de um arquivo de áudio (Jazz3.wav) com uma simulação de perda de pacotes em torno de 10%. Pôde-se claramente perceber que devido ao fator perdas ser elevado a política de adaptação foi aplicada e resultou em uma modificação significativa de largura de banda utilizada pela aplicação. O formato inicial do áudio escolhido para transmissão foi dvi/rtp, o qual consumiu cerca de 88 Kbps de largura de banda da rede, sendo modificado posteriormente para ULAW/rtp (64 Kbps). Isso depois do primeiro *report* do receptor ter informado perda de pacotes em 10% (tempo 6s de transmissão). Posteriormente, o áudio foi trocado para dvi/rtp (32 Kbps) e gsm/rtp (16 Kbps), tempos 27s e 48s respectivamente. A transmissão do mesmo arquivo com simulação de perdas variável é ilustrada na Figura 9. Nesta transmissão (Figura 9), o formato do *stream* foi modificado duas vezes; na primeira o *stream* foi modificado para ULAW/rtp (64Kbps) e na segunda para dvi/rtp (32 Kbps), tempo 10s e 45s respectivamente.

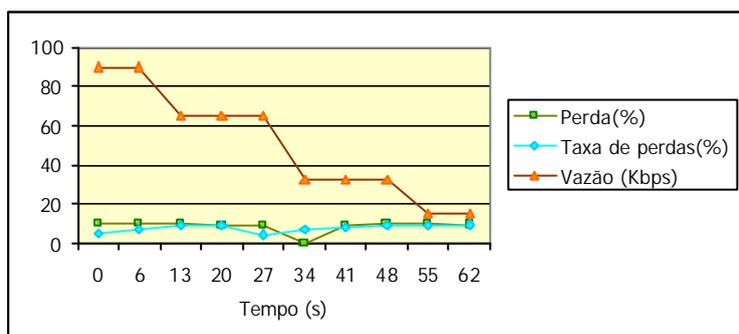


Figura 8 - Transmissão de áudio (Jazz3.wav) com simulação de 10 % de perdas

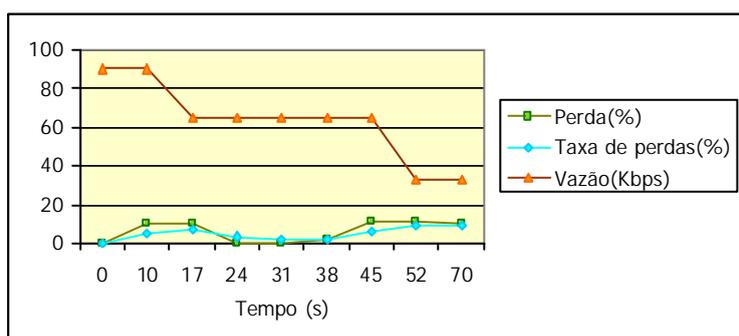


Figura 9 - Transmissão de áudio (Jazz3.wav) com simulação de perdas variável

O gráfico representado pela Figura 10 ilustra a transmissão de um arquivo de áudio (Trailer.mov) com taxa de perdas em torno de 10%. A política de adaptação foi aplicada duas vezes (tempo 6s e tempo 32s). Na primeira, o formato do *stream* foi modificado para dvi/rtp com 32 Kbps de vazão e posteriormente para gsm/rtp com vazão de 16 Kbps. O áudio inicial de transmissão possuía formato dvi/rtp (44 Kbps). Na Figura 11, o mesmo arquivo foi transmitido, porém com simulação de perda de pacotes variável, resultando na aplicação da política de

adaptação duas vezes (tempos 31s e 59s). Os formatos modificados foram os mesmos da transmissão do arquivo (Trailer.mov) com percentual de perdas em 10%.

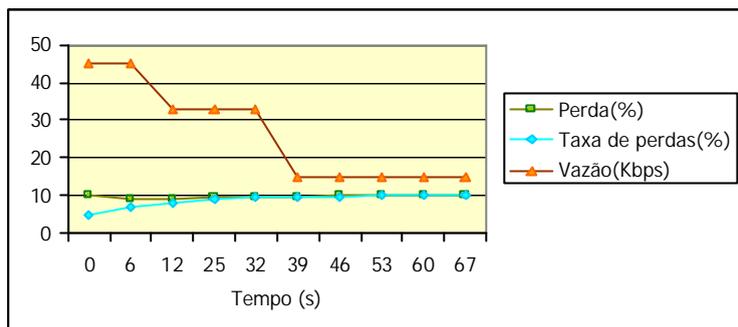


Figura 10 - Aplicação de política de adaptação com 10% de perdas

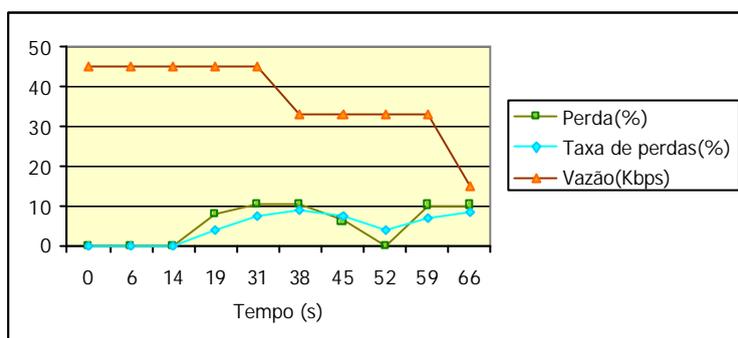


Figura 11 - Aplicação de política de adaptação com perda de pacotes variável

Vários testes foram realizados com outros arquivos de áudio com diferentes formatos. Com isso, observou-se que em cenários com perdas de pacotes excessivas, a qualidade da mídia melhorou no receptor quando a política de adaptação em questão foi aplicada. O áudio passou gradativamente de um "estado" cuja audição era truncada e incompreensível para outro no qual mostrou-se compreensível. Além de melhorar a compreensão, modificar a taxa de transmissão pode ajudar a reduzir o congestionamento da rede (a exemplo do protocolo TCP) através da diminuição do volume dos dados a serem transmitidos. Na maioria dos testes realizados, a troca de um formato por outro durante a transmissão resultou em uma diminuição de largura de banda utilizada pela aplicação em aproximadamente 4 à 5 vezes em relação ao volume inicial do formato escolhido.

6 Considerações Finais

Frente a realidade atual do modelo de serviços oferecidos pela Internet, muitas aplicações multimídia que transmitem ou recebem áudio e vídeo necessitam tornarem-se adaptativas às condições que a rede oferece. Consequentemente, essas aplicações requerem que suporte adicional seja disponibilizado para que obtenham garantias de QoS. Contudo, oferecer garantias de QoS em cenário adaptativo, como a Internet, requer habilidade do desenvolvedor da aplicação para fazer isso e nem sempre ele está preparado para realizar tal tarefa, pois essa implica no tratamento de detalhes de comunicação como atraso, perda de pacotes, variação de atraso e largura de banda utilizada pela aplicação.

Deste modo, este trabalho apresentou uma Camada de Adaptação de QoS que possui mecanismos de monitoração e adaptação de QoS. A monitoração dá-se através da análise dos

pacotes RTCP *Sender Report* e RTCP *Receiver Report*, pois ambos permitem monitorar a qualidade da distribuição dos dados. Adaptação, como segundo mecanismo, fornece meios da aplicação ajustar-se as variações dinâmicas que acontecem na rede de comunicação através de um conjunto de políticas adaptativas, as quais podem ser escolhidas pelo desenvolvedor da aplicação. Contudo, para o efetivo desenvolvimento desses mecanismos alguns aspectos principais foram levados em consideração, como: (i) a necessidade de obtenção e tratamento contínuo dos dados obtidos da rede; (ii) a necessidade de modificações no comportamento da aplicação segundo análise dos dados coletados; (iii) a necessidade dos dois processos acima acontecerem de maneira concorrente com o envio e a recepção dos dados.

Isso não é uma tarefa fácil de ser realizada, principalmente considerando-se algumas questões, como por exemplo, como obter os dados referentes ao tráfego da rede? Depois, como tratar esses dados e o que fazer com eles? Finalmente, como modificar o comportamento da aplicação frente aos dados coletados e analisados? Questões como essas foram criteriosamente analisadas e observadas para que o trabalho produzisse os resultados esperados.

Uma Interface para Programas de Aplicação (API) foi desenvolvida para oferecer os serviços de monitoração e adaptação de QoS e visa ajudar a tarefa do desenvolvedor de uma aplicação multimídia, escrita na linguagem de programação Java, a incorporar QoS à aplicação. Ela foi projetada para ser fácil de programar, tornar possível que desenvolvedores implementem aplicações multimídia com suporte à QoS baseadas na API existente e também integrem novas características a estrutura oferecida. A API oferece uma solução básica para uma classe de problemas em que desenvolvedores podem explorar sua estrutura através da extensão. Além disso, as funcionalidades oferecidas também podem ser facilmente customizadas.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] BLAKE, S., et al. "An Architecture for Differentiated Services", Internet RFC 2475, **Internet Engineering Task Force**, Dec. 1998.
- [2] BOLOT, J., TURLETTI. "A rate control mechanism for a packet video in the Internet", **Proceedings IEEE INFOCOM'94**, p.1216-1223, June 1994.
- [3] BRADEN B., ZHANG, L., BERSON, S., HERZOG, S. and JAMIN, S. "Resource ReSerVation Protocol (RSVP), Internet RFC 2205, **Internet Engineering Task Force**, Oct. 1997.
- [4] BUSSE, Ingo, DEFFNER, B., SCHULZRINNE, H. "Dynamic QoS Control of Multimedia Applications based on RTP", **Computer Communications**, [s.l.], vol 19, p. 49-58, Jan. 1996.
- [5] CAMPBELL, Andrew, COULSON, G., HUTCHISON, D. "Flow Management in a Quality of Service Architecture", In: IFIP INTERNATIONAL CONFERENCE ON HIGH PERFORMANCE NETWORKING, 5, 1994, Grenoble. **Proceedings...** Grenoble, 1994.
- [6] FRY, M., SENEVIRATNE, A. VOGEL, A. WITANA, V. "Delivering QoS Controlled Continuous Media on the World Wide Web", In: Internacional Workshop on Quality of Service (IWQoS'96), 4, 1996, Paris. **Proceedings...** Paris, 1996. P. 115-126.
- [7] GROSS, Thomas., BOLLIGER, J. "A Framework-Based Approach to the Development of Network-Aware Applications", **IEEE Transactions on Software Engineering**, vol. 24, nº 5, p. 376-389, May 1998.
- [8] HUTCHISON, D. EL-MARAKBY, R. "Towards Managed Real-time Communication in the Internet Environment", In: Workshop on the Architecture and Implementation of High Performance Communication Systems (HPCS '97), 4, 1997, Greece. **Proceedings...** Greece, June 1997.

- [9] HUARD, J. F., INOUE, I., LAZAR, A. A., YAMANAKA, H. "Meeting QoS Guarantees by End-to-End QoS Monitoring and Adaptation", In: Workshop on Multimedia and Collaborative Environments of the Fifth IEEE International Symposium On High Performance Distributed Computing (HPDC-5), 5, 1996, Syracuse NY. **Proceedings...** Syracuse NY, Aug. 1996.
- [10] JMF, "Java Media *Framework* 2.0 Guide". API Documentation. **Sun Microsystems**, March 1999. Capturado em 16 de fevereiro 2000. Online. Disponível na Internet <http://www.java.sun.com/products/java-media/jmf/index.html>
- [11] LUNARDI, Sediane C. **Uma Camada de Suporte à Qualidade de Serviço para Aplicações Multimídia na Internet**, Porto Alegre, 2001. Dissertação (Mestrado em Ciência da Computação) — Faculdade de Informática, PUCRS, 2001.
- [12] MARAKBY, R., HUTCHISON, D. "Delivery of Real-time Continuous Media over the Internet", Internet Report, Lancaster University, 1997. Capturado em 23 de outubro 1999. Online. Disponível na Internet http://www.comp.lancs.ac.uk/computing/research/mpg/index_qosm.html
- [13] McCANNE, Steven. "Scalable Multimedia Communications Using IP *Multicast* and *Lightweight Session*", **IEEE Internet Computing**, p.33-45, March/April 1999.
- [14] MOGHE, P., KALAVADE, A. "Terminal QoS of Adaptive Applications and its Analytical Computation". In: Internacional Workshop on Quality of Service (IWQoS'97), 5, 1997, New York – USA. **Proceedings...** New York, 1997. p.369-380.
- [15] SCHULZRINNE, Henning. "Audio and Video Over Packet Networks — Issues, Architecture and Protocols", **Interop'94**, Paris, France, 1994.
- [16] SCHULZRINNE, Henning, et al. "RTP: A Transport Protocol for Realtime Applications", Internet RFC 1889, **Internet Engineering Task Force**, Jan. 1996.
- [17] SCHULZRINNE, Henning. "RTP Profile for Audio and Video Conferences with Minimal Control", Internet RFC 1890, **Internet Engineering Task Force**, Jan. 1996.
- [18] SISALEM, Dorgham. "End-to-End Quality of Service Control Using Adaptive Applications", In: Internacional Workshop on Quality of Service (IWQoS'97), 5, 1997, New York - USA. **Proceedings...** New York, 1997. p.381-395
- [19] SISALEM, Dorgham, SCHULZRINNE, H. "The Multimedia Internet Terminal (MINT)". **Telecommunications Systems**, vol. 9, n° 3-4, p. 423-444, Sept. 1998.
- [20] TANENBAUM, Andrew. **Computer Networks**. Englewood Cliffs, NJ : Prentice Hall, 1996. 3^a ed. 658p.
- [21] VEGA-GARCÍA, Andrés, BOLOT, J. "Control Mechanisms for Packet Audio in the Internet". **Proceedings IEEE INFOCOM'96**, San Francisco, California, p.232-239, April 1996.
- [22] VOGEL, L., KERHERVÉ, B., BOCHMANN, G., GECSE, J. "Distributed Multimedia and QoS: A Survey", **IEEE Multimedia**, p. 10-19, Summer 1995.
- [23] WADDINGTON, D. and HUTCHISON, D. "A General Model for QoS Adaptation", In: Internacional Workshop on Quality of Service (IWQoS'98), 6, 1998, California. **Proceedings...** Napa, California, 1998. p. 275-277.