

# SAM – Sistema Adaptativo para Multimídia

Helena L. Braga<sup>1</sup>, Maiko de Andrade<sup>1</sup>, José V. de Lima<sup>2</sup>, Valter Roesler<sup>1,2</sup>

<sup>1</sup>UNISINOS – Universidade do Vale do Rio dos Sinos  
Avenida Unisinos, 950 – São Leopoldo – RS

<sup>2</sup>Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)  
Caixa Postal 15064 - Av. Bento Gonçalves, 9500 Bl. IV – Porto Alegre – RS

{helenal, maiko, roesler}@exatas.unisinos.br, valdeni@inf.ufrgs.br

**Resumo.** Este artigo descreve uma ferramenta a ser utilizada para assistir transmissões multimídia ao vivo, como aulas a distância, shows e canais de TV. Seu maior benefício é aumentar o desempenho e a acessibilidade da transmissão, pois o sinal é codificado e enviado em camadas ao receptor, que utilizando a ferramenta, adapta-se de acordo com a sua capacidade de rede no momento.

## Introdução

Um dos grandes desafios da Internet atualmente é permitir acesso universal às transmissões multimídia, como, por exemplo, uma conferência em tempo real na Internet, sendo transmitida simultaneamente para diversos usuários em redes com diferentes larguras de banda e com vários tipos de resolução de tela, ou mesmo características de processamento muito diferentes.

Uma possível solução seria a codificação do sinal e transmissão em *multicast* que permite que uma única transmissão seja entregue a um grande número de receptores. No entanto, devido à heterogeneidade da rede (alguns usuários localizados em enlaces lentos, outros em enlaces rápidos), o uso de uma única codificação torna-se um problema, exigindo que este tipo de solução utilize um método que possibilite acesso tanto aos receptores localizados em enlaces rápidos quanto aos localizados em enlaces lentos.

Uma alternativa para a solução da heterogeneidade da rede é a transmissão do sinal multimídia em camadas, na qual o fluxo multimídia é codificado em várias camadas no transmissor e cada camada é transmitida como um grupo *multicast* diferente. Cada camada adiciona qualidade à anterior e o receptor inscreve-se em tantas camadas quanto a sua condição de rede e de máquina permitirem. Esta solução foi definida por McCanne em [MCC 96]. Há também a possibilidade de existirem “camadas opcionais”, que são camadas que são adequadas a apenas um grupo de receptores, como, por exemplo, dublagem de voz em linguagens diferentes.

A ferramenta mostrada neste artigo utiliza o mesmo conceito proposto por McCanne, no entanto, a forma de adaptação realizada no receptor é completamente diferente. O método implementado é baseado no algoritmo ALM (*Adaptive Layered Multicast*), descrito detalhadamente em [ROE 02]. Mesmo estando em um estágio inicial de desenvolvimento, a ferramenta tem mostrado que o algoritmo se adapta corretamente tanto em redes de alto desempenho como em redes de menor desempenho.

## Visão Geral do Sistema

O transmissor deve fornecer a codificação do sinal multimídia em camadas e transmitir cada camada como um grupo *multicast* diferente. No lado do receptor, a adaptação é baseada em detecção de perdas e estimativa de equidade de banda. Este sistema é descrito na figura 1. Em um primeiro momento o receptor obtém os parâmetros da sessão a partir de um arquivo de configuração. Este arquivo contém as características de cada camada, como o endereço IP multicast, a porta e a taxa de transmissão. A partir das informações obtidas inicialmente, o receptor inscreve-se no número correto de camadas de acordo com a capacidade da rede e seu nível de congestionamento, sempre buscando justiça em relação aos demais receptores e tráfegos (efetuado através do controle de camadas do ALM). Após o recebimento das camadas, estas são enviadas ao módulo de decodificação, que reconstrói e reconstitui o sinal multimídia, e este sinal é exibido ao usuário.

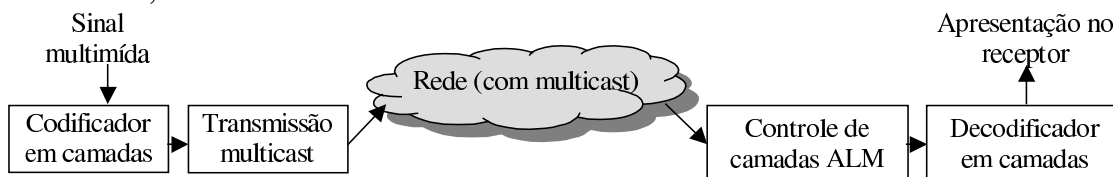


Figura 1. Visão Geral do Sistema.

Como visto anteriormente, o protocolo ALM proposto por Roesler em [ROE 02] utiliza o conceito da transmissão em camadas, sendo composto de duas fases: “start-state” e “steady-state”. A fase “start-state” tenta alcançar a sua banda equitativa com os fluxos concorrentes incrementando rapidamente a banda até a detecção de perdas. Essa fase pode ser vista no gráfico da figura 3, que mostra uma subida rápida até a quinta camada até a detecção de perdas e depois uma queda para a quarta camada, e então se estabilizando, marcando a entrada do ALM na fase “steady-state”. Na fase “steady-state”, o incremento da banda é mais lento. Isso também pode ser visto no gráfico da figura 3, onde o algoritmo demora aproximadamente 300 segundos para tentar subir para a quinta camada. Nesse caso a rede não suportou o aumento de banda, gerando perdas e fazendo com que o receptor deixe a camada mais alta, como mostra a figura.

## Descrição da Implementação

Existem atualmente dois aplicativos desenvolvidos: um transmissor CBR de *multicast* em camadas e um receptor que executa o algoritmo do ALM, adaptando-se conforme explicado em [ROE 02].

Inicialmente o transmissor lê um arquivo de configuração que contém uma linha descrevendo cada camada. As características descritas são: IP *multicast* da camada, porta destino a ser enviado o pacote, taxa em kbits/s e tamanho do pacote em bytes. A partir desta configuração lida, é criada uma *thread* para a transmissão de cada camada. No momento de transmitir um determinado pacote, a *thread* adiciona um número de sequência e *timestamp* ao pacote. A partir da taxa de cada camada é calculado um tempo de espera entre a transmissão de dois pacotes sucessivos, e o algoritmo mantém-se assim indefinidamente.

Atualmente o transmissor fica em laço enviando um arquivo qualquer, visto que o codificador em camadas encontra-se em desenvolvimento através de uma dissertação de mestrado.

O receptor lê o mesmo arquivo de configuração lido pelo transmissor, obtendo a banda de cada camada e a banda necessária para se inscrever em uma nova camada (grupo *multicast*). No receptor, além das *threads* que são criadas por camada, também é criada uma thread de controle. O receptor inscreve-se em novas camadas conforme o módulo de controle permitir. Após a inscrição em uma nova camada o receptor passa a receber os dados do grupo *multicast* correspondente, ou deixa esta camada, de acordo com a necessidade, parando de receber o tráfego associado.

O ambiente controlado onde os testes foram realizados pode ser visto na figura 2. Cada transmissor envia 6 camadas distintas que crescem de forma exponencial. O transmissor *sender1* envia as camadas 239.1.1.1 a 239.1.1.6, e o transmissor *sender2* envia as camadas 239.1.1.11 a 239.1.1.16.

Estes transmissores são ligados a um *switch* e este é ligado a um roteador. Na outra sub-rede ficam os receptores (ligados a outro *switch*). Nos testes realizados o roteador limitava o tráfego a 1Mbit/s. A versão do *mrouted* foi a 3.9 beta 3, alterado por Rizzo [RIZ 98] para menor tempo de *leave*. Os receptores entram e saem dos grupos *multicast* utilizando o protocolo IGMP (*Internet Group Management Protocol*) padrão.

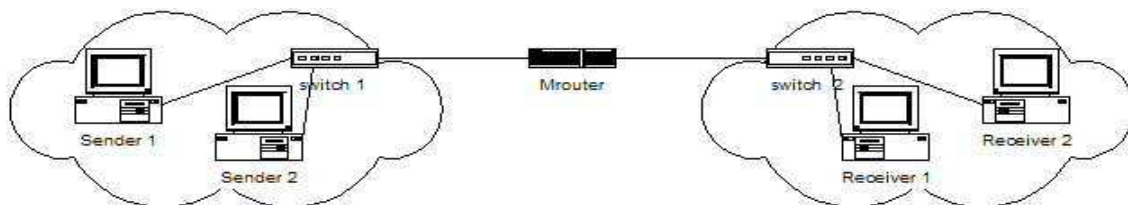


Figura 2. Ambiente de teste.

Diversos testes foram realizados nesse ambiente controlado, com variações no número de camadas, taxa de cada camada e número de receptores (1 e 2). Os resultados mostraram um comportamento semelhante ao visto na figura 3, onde é apresentada a interface do sistema SAM, que mostra um dos dois receptores concorrentes. Esse ambiente foi desenvolvido com o propósito de visualizar algumas das variáveis do sistema e ter uma noção geral do que está acontecendo durante os testes. A parte superior esquerda da figura 3 mostra as camadas existentes e suas respectivas taxas de transmissão (*rate*), o IP *multicast* da camada (*IP Address*), os pacotes perdidos nos últimos 5 segundos (*Lost 5s*) e todos os pacotes perdidos de cada camada (*Lost tot*).

No lado superior direito da figura 3 são mostrados os valores de algumas variáveis do sistema como: o tempo em segundos em que o sistema ficará rodando (*Execute* – que roda indefinidamente caso seja zero), a quantidade de banda que o algoritmo está autorizado a usar (*Bwshare*), a banda que o algoritmo está usando (*Bwused*), a soma dos pacotes perdidos de todas as camadas (*Total Lost*), o intervalo de execução em milissegundos (*EI*) e o tempo em segundos decorrido desde que foi iniciado o algoritmo (*Time*).

O sistema conta também com um gráfico de tempo referente à camada em que o sistema se encontra em um dado momento.

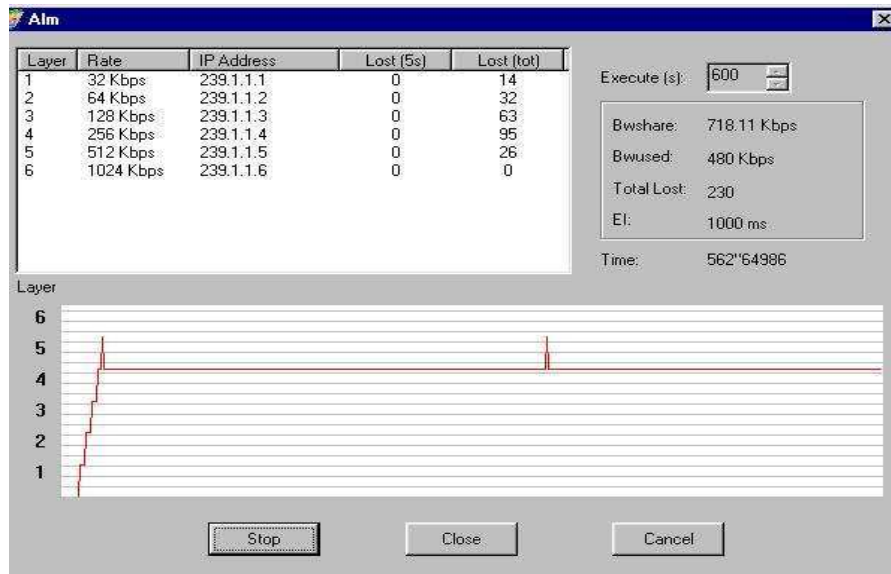


Figura 3. Interface com o usuário – *receiver1* associado ao *sender1*.

## Conclusões

A implementação está se mostrando coerente com o algoritmo proposto, porém, ainda faltam alguns testes para que se obtenha uma ferramenta utilizável em Ensino a Distância. É necessário, entre outras coisas, a obtenção do codificador / decodificador em camadas, a análise do funcionamento numa rede não controlada, testes com *unicast*, análise do nível de perdas e verificação da possibilidade de inserir redundância suficiente na transmissão para uma recepção com qualidade aceitável.

Entretanto, os testes efetuados até agora mostraram que o sistema possui grande potencial de utilização no futuro, viabilizando o acesso às transmissões ao vivo por pessoas localizadas tanto em redes de alto desempenho como em redes de menor desempenho, aumentando progressivamente o tráfego conforme o desempenho e nível de congestionamento do receptor.

## Referências

- [MCC 96] McCANNE, S.; JACOBSON, V.; VETTERLI, M. **Receiver driven layered multicast**. In: ACM SIGCOMM 96. Proceedings...Stanford, CA, pp 117-130, Agosto 1996.
- [RIZ 98] RIZZO, L. **Fast Group Management in IGMP**. In: Hipparch Workshop, London, June 1998.
- [ROE 02] ROESLER, V.; BRUNO, G.G.; LIMA, V. **Análise de estabilidade em um algoritmo para controle de congestionamento de transmissões multimídia em camadas**. In: VIII SIMPÓSIO BRASILEIRO DE MULTIMÍDIA E HIPERMÍDIA, SBMIDIA, 2002. Anais... Fortaleza: Universidade Federal do Ceará, Outubro 2002.