

Um Protocolo de Multicast Semi-confiável para Aplicações Multimídia Distribuídas em Redes de Larga Escala*

Christiane Montenegro Bortoleto¹, Lau Cheuk Lung¹, Frank A. Siqueira²,
Alysson Neves Bessani³, Joni da Silva Fraga³

¹Programa de Pós-Graduação em Informática Aplicada
Pontifícia Universidade Católica do Paraná (PUC-PR)

Rua Imaculada Conceição, 1155 Cep 80.215-901 – Curitiba – PR – Brasil

²Departamento de Informática e Estatística, ³Departamento de Automação e Sistemas
Universidade Federal de Santa Catarina (UFSC)

Campus Universitário, Caixa Postal 476 – CEP 88040-900 – Florianópolis – SC – Brasil

chrisb@das.ufsc.br, lau@ppgia.pucpr.br, frank@inf.ufsc.br,
neves@das.ufsc.br, fraga@das.ufsc.br

Abstract. *This paper proposes a semi-reliable multicast protocol that aims to increase the quality of video streams transmitted in large-scale systems without overloading the video source and the communications network. This protocol, which is based on the IP multicast protocol and the MPEG standard, evaluates the necessity of retransmitting lost packets taking into account the capacity of the corresponding MPEG frames to improve the quality of the video stream. The proposed protocol relies on the neighboring receivers for retransmitting lost packets, resulting in much faster recovery, which is vital in order to receive retransmitted packets on time to be exhibited. Besides, this strategy avoids overloading the video source, making it more scalable than the traditional approach of retransmitting from the source. Simulation results presented in this paper show that the proposed protocol delivers video streams to the receivers with a better image quality.*

Resumo. *Este artigo apresenta a proposta de um protocolo multicast semi-confiável que objetiva incrementar a qualidade de fluxos (streams) de vídeo transmitidos em sistemas de larga escala sem sobrecarregar a fonte do vídeo e a rede de comunicação. Empregando o protocolo de comunicação Multicast IP e tendo como base a hierarquia de quadros do padrão MPEG, este protocolo avalia a necessidade de efetuar a retransmissão de pacotes perdidos com base na capacidade do quadro MPEG correspondente em incrementar a qualidade do fluxo de vídeo. O protocolo proposto emprega os receptores vizinhos para efetuar a retransmissão, permitindo que os quadros perdidos sejam recuperados mais rapidamente, o que é essencial para que estes sejam recebidos em tempo para serem exibidos. Além disso, esta estratégia evita sobrecarregar a fonte do fluxo de vídeo, tornando o protocolo mais escalável que as abordagens tradicionais que realizam a retransmissão a partir da origem. Os resultados das simulações apresentadas neste artigo mostram que o protocolo proposto provê uma melhor qualidade de imagem aos receptores de fluxos de vídeo.*

* Parcialmente financiado pelo CT-Info/MCT/CNPq – Projeto: 506639/2004-5 e FunPesquisa 200/PRPG-UFSC.

1. Introdução

Aplicações multimídia em tempo real estão sendo intensamente desenvolvidas em diversas plataformas e topologias. Hoje em dia encontramos comumente aplicações multimídia em computadores pessoais e em pequenos dispositivos de comunicação. As estruturas para Internet e intranet estão utilizando cada vez mais serviços de rede integrados, onde aplicações de características diferenciadas são executadas usando um mesmo modelo de serviços e estrutura de rede. No entanto, a estrutura de comunicação comum oferece riscos à qualidade do conteúdo multimídia em função das políticas de transmissão, roteamento (descarte de pacotes) e da complexidade na formatação dos dados. A infra-estrutura de redes necessita de mecanismos para aperfeiçoar e garantir o desempenho dessas aplicações. A limitação física em caso de sobrecarga nos roteadores, para direcionamento e retransmissão de datagramas, gera perda de informações para as aplicações de tempo real, prejudicando a qualidade do conteúdo apresentado.

As aplicações multimídia distribuídas possuem requisitos bastante peculiares que não se encontram presentes em outras aplicações distribuídas. Um fluxo (*stream*) de áudio/vídeo, por exemplo, requer que os dados sejam recebidos no momento certo e é complacente com dados perdidos. Se um pacote chega após um determinado prazo, este não contribui em nada na reprodução, ou seja, o efeito observável seria o mesmo que se o pacote tivesse sido perdido. Pequenas perdas no *stream* de áudio/vídeo podem ser substituídas ou ocultadas sem causar uma perda de qualidade de reprodução perceptível para o usuário. Em particular, nas videoconferências realizadas atualmente através da Internet, o suporte de comunicação é a principal causa pelo baixo desempenho.

A difusão de mídia em um grupo de processos é um dos alicerces sobre os quais se fundamentam a televisão digital e outras tecnologias de difusão de multimídia. Estas novas tecnologias ainda carecem de pesquisa e são exemplos representativos de como as técnicas criadas inicialmente dentro do contexto da computação podem ser aplicadas em áreas tão diversas como a difusão de mídia. Abstrações de *multicast* têm recebido especial interesse da comunidade [Barcellos e Roesler, 2000] pela facilidade e por apresentar um desempenho melhor que comunicações ponto a ponto. Diferentes aplicações distribuídas, que fazem uso de mecanismos de *multicast*, têm requisitos muito diferentes no que diz respeito à confiabilidade na entrega de pacotes. Por exemplo, algumas precisam de ordenamento total de mensagens, outras trocam a confiabilidade total por um melhor tempo de entrega e outras podem precisar ter mais de um membro como fonte de dados [Hadzilacos, 1994]. Estes diversos requisitos têm diferentes impactos no projeto dos protocolos de *multicast*.

Na literatura, existe uma quantidade significativa de trabalhos em *multicast* melhor esforço (*best-effort*) para aplicações multimídia distribuída [Barcellos, 1998] e *multicast* confiável [Hadzilacos e Toueg, 1994] para aplicações que impõem confiabilidade na entrega das mensagens (ex: aplicações tolerantes a faltas). No entanto, entre esses dois extremos, encontramos os protocolos de *multicast* semi-confiável, um tema de pesquisa bem mais recente e ainda pouco explorado [Piecuch et al., 2000; Mane, 2000; Pereira et al., 2003; Xu et al., 1997; Schneyer et al., 1999]. *Multicast semi-confiável* é uma abstração de comunicação em que os dados (pacotes) são qualificados, usualmente pela aplicação, antes da sua transmissão, estabelecendo níveis de importância ou prioridade na correção de erros (retransmissão).

Este artigo propõe um protocolo de difusão semi-confiável mais eficiente e escalável, projetado para aplicações multimídia distribuída baseada em grupo (ex: videoconferência, difusão de vídeo digital, etc). Será mostrado, através de simulações, que

este protocolo é mais apropriado para redes de larga escala, tal como a Internet. O protocolo é especificado para difusão de fluxos de vídeo empregando codificadores que apresentem alguma forma de hierarquização de quadros, tal como no padrão MPEG, permitindo, desta forma, estabelecer níveis de prioridade na correção de quadros perdidos.

O artigo apresenta na seção 2 as características do padrão MPEG. Na seção 3, são mostrados os conceitos relevantes acerca de *multicast* semi-confiável. O protocolo proposto é apresentado na seção 4. Dados experimentais são apresentados na seção 5. A seção 6 cita alguns trabalhos relacionados e, finalmente, na seção 7 são apresentadas as conclusões finais e perspectivas para o futuro deste trabalho.

2. O Padrão MPEG

Existem muitos esquemas de codificação tanto para áudio quanto para vídeo, com diferentes características de desempenho. Os esquemas de codificação que requerem pouca largura de banda freqüentemente resultam em uma reconstrução menos precisa do sinal analógico original. Há também uma diferença na complexidade e tempo de processamento necessário entre os esquemas.

Algumas mídias, como vídeo, ainda têm que ser comprimidas antes da transmissão. Sem a compressão, o vídeo pode precisar de mais de 400 Mbps, o que ainda é inviável nas redes atuais. Os padrões mais importantes para compressão de vídeo são da família MPEG (*Motion Picture Experts Group*) [Chiariglione, 2000]. Existem vários padrões diferentes, entre eles o MPEG-2 que resulta em taxas de compressão de 3 Mbps a 100 Mbps. O algoritmo de compressão do MPEG-2 baseia-se na correlação entre *pixels* e na correlação de movimento translacional entre quadros consecutivos. Como a maioria dos quadros de uma seqüência de imagens é muito parecida, exceto por deslocamentos devido a movimentos, é possível codificar um quadro através do cálculo do vetor de deslocamento em relação à imagem anterior [Schneyer et al., 1999]. O *stream* de saída consiste em três tipos de quadros:

- Quadros I (*Intracoded*): imagens completas codificadas individualmente;
- Quadros P (*Predictive*): quadros codificados com predição relativa ao último quadro;
- Quadros B (*Bidirectional*): diferenças entre o último e o próximo quadro.

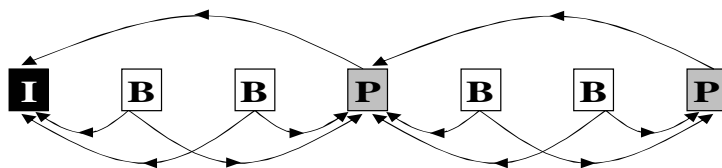


Figura 1. Relacionamento de Quadros IPB.

Os quadros I são inseridos no *stream* de saída a uma taxa específica (por exemplo, uma ou duas vezes por segundo), com quadros P e B entre eles. Os quadros I não são dependentes de outros quadros para serem decodificados, mas são necessários para decodificação de quadros P e B. Os quadros P são necessários para decodificação de quadros do tipo B e são baseados em previsão antecipada (do tipo 'P' - *forward prediction*) que são previstos pelo último quadro de referência, que pode ser um quadro do tipo 'I' ou 'P'. O quadro do tipo 'P' possui a referência do vetor de movimento no passado, utilizado para referenciar o bloco do quadro anterior na mesma posição do quadro atual. Quadros do tipo 'B' (*backward prediction*) são previstos baseados em referências de quadros anteriores e próximos, que podem ser do tipo 'I' ou 'P'. O quadro do tipo 'B' possui a referência do vetor de movimento no futuro, utilizado para referenciar o bloco do próximo quadro na

mesma posição do quadro atual. Este método de codificação faz com que alguns quadros sejam mais importantes que os outros. Se um quadro I é perdido durante a transmissão, não será possível decodificar os quadros que cheguem antes do próximo quadro I. A relação entre os quadros em um fluxo de vídeo MPEG é ilustrada na Figura 1. Cada GOP (*Group Of Pictures*) deve conter ao menos um quadro do tipo 'I'. Um grupo de figuras (GOP) é uma série de quadros onde cada um consiste em um cabeçalho da figura e os seus atuais dados. A sequência de um grupo de figuras (GOPs), é uma sequência de quadros entre quadros consecutivos do tipo 'I' de tamanho N.

Considerando-se estas características do MPEG, é possível escolher o método mais adequado para transmissão *multicast* dos pacotes contendo os quadros do padrão MPEG. A seção seguinte descreve a idéia central dos protocolos de *multicast* semi-confiável, ilustrando sua adequação para transmissão multimídia.

3. Multicast Semi-confiável

Como as aplicações multimídia distribuída têm requisitos bastante peculiares em relação às demais, é conveniente utilizar os protocolos de *multicast* semi-confiável [Piecuch et al., 2000, Mane, 2000, Pereira et al., 2003] que buscam a garantia relaxada de entrega dos pacotes, desde que respeitadas as limitações impostas pela aplicação.

Multicast semi-confiável é uma abstração de comunicação em que nem todos os pacotes são necessariamente retransmitidos quando solicitados, é dada uma prioridade para aqueles pacotes que forem de vital importância para manter a qualidade da informação requerida pela aplicação. Isto é, dado um conjunto de pacotes a ser difundido a um grupo de receptores a entrega confiável é garantida apenas a um subconjunto (os de maior prioridade) deste conjunto. Um pacote difundido de forma confiável significa que ele terá que ser entregue por todos receptores corretos (propriedade de acordo [Hadzilacos e Toueg, 1994]). Essa garantia pode ser obtida usando algum mecanismo de correção de erro (ex: retransmissão de pacotes). Nos pacotes restantes do conjunto (os de menor prioridade), erros de transmissão só serão corrigidos se as condições da rede (ex: tráfego, congestionamento, latência, etc) assim o permitirem. Caso contrário, a entrega é baseada no melhor esforço (*best-effort*) em que a propriedade de acordo pode ser violada neste caso.

Os pacotes (ou mensagens) a serem difundidos a um grupo de receptores podem ser qualificados dentro de uma hierarquia, baseado em alguma semântica da aplicação [Pereira et al., 2003] ou alguma regra do próprio protocolo de difusão semi-confiável, estabelecendo níveis de importância ou prioridade na correção de erros (retransmissão seletiva [Piecuch et al., 2000, Mane, 2000]). Portanto, a correção de erro é efetuada baseada nessa prioridade e em parâmetros que reflitam o estado de rede (ex: tráfego, congestionamento, latência, etc). A adoção desta estratégia permite que sejam tomadas decisões pró-ativas [Martins e Farines, 2003] para efetuar a recuperação de pacotes em caso de perda.

3.1 Multicast Semi-confiável para Aplicações Multimídia Distribuídas

A transmissão de informações multimídia tem uma taxa fixa e os quadros devem ser recebidos e reproduzidos no receptor a uma taxa similar para manter o significado original da sequência. Assim, cada unidade de informação enviada deve ser recebida dentro de determinado intervalo de tempo. Além disso, a fração de unidades de dados perdida (ou entregue após o instante no qual deveria ser exibida) deve estar também dentro de um limite aceitável. Portanto, o problema de QoS envolve encontrar os limites através das redes para erros de transmissão e possíveis flutuações entre os intervalos de entrega.

Um mecanismo de *multicast* confiável tradicional não é adequado para este tipo de aplicação, por várias razões. Primeiramente, a estratégia de retransmissão com *timeout* obtém confiabilidade às custas do aumento de latência. Em segundo lugar, aplicações multimídia podem tolerar erros provenientes da perda e corrupção de pacotes dentro de um limite aceitável. Assim, um protocolo de transporte multimídia exige uma entrega semi-confiável dos pacotes, onde o atraso é mais relevante do que a entrega de todos os pacotes de um conjunto.

Outros tipos de aplicação também podem fazer uso do *multicast* semi-confiável para melhoria de seu desempenho. Em [Pereira et al., 2003], o autor cita o exemplo de jogos *multiplayer* distribuídos, onde mensagens obsoletas podem ser eliminadas sem prejudicar o resultado final.

4. Protocolo Proposto

O protocolo tira proveito da hierarquia de quadros definida no padrão MPEG em protocolos de difusão semi-confiável através da qualificação dos quadros MPEG e do encapsulamento desses em pacotes UDP. Uma vez qualificados, a correção de erros (através da retransmissão dos pacotes perdidos) pode ser realizada pelo emissor (fonte) ou pelos receptores de acordo com o tipo de quadro perdido (I, P ou B) e de acordo, por exemplo, com os parâmetros da rede (tráfego, taxa de recepção, congestionamento, etc). A idéia inicial é que todos os quadros do tipo I sejam confiavelmente entregues e quadros do tipo P e B sejam, principalmente o primeiro, retransmitidos (em caso de perda) dependendo das condições do meio de transmissão. Este protocolo está baseado nas idéias do protocolo ReMIOP [Bessani et al., 2003] e em alguns conceitos apresentados em [Martins e Farines., 2003].

O algoritmo apresentado a seguir descreve os procedimentos para recepção e transmissão das mensagens no protocolo. Neste algoritmo, considera-se que ao detectar um pacote perdido, o emissor consiga saber qual o tipo de quadro estava contido neste pacote.

4.1. Funcionamento do Protocolo

Os quadros (pacotes) são difundidos pelo emissor diretamente no grupo (linha 1), sem que haja conhecimento prévio de seus membros. Os receptores adicionam todos os pacotes recebidos pela primeira vez a um *buffer* e entregam o quadro à aplicação (linhas 5-8). Note que na linha 6 o receptor cancela qualquer possível pedido de retransmissão da mensagem recebida que ele estava prestes a enviar. Em seguida, eles detectam perdas de pacotes (linha 12) através da busca por lacunas na sequência dos pacotes recebidos, que é explicitada através de um número de ordem no cabeçalho de cada pacote. Quando um receptor detecta uma lacuna, ele avalia a necessidade de pedir a retransmissão do(s) pacote(s) perdido(s), através de uma mensagem NACK (linhas 13,18 e 19). Esta avaliação é feita com base nos requisitos de QoS da aplicação e nos parâmetros de rede. A QoS da aplicação leva em conta a relevância dos quadros perdidos para a reprodução da mídia no receptor e se o quadro quando retransmitido ainda será útil no momento em que chegar. Os parâmetros de rede levam em conta a taxa de perda de pacotes, seu congestionamento e o atraso aceitável.

Se um quadro perdido for considerado relevante (linhas 16-19), a retransmissão é solicitada através da difusão de uma mensagem de controle (NACK) no grupo (linhas 21), caso contrário, o receptor simplesmente ignora o pacote perdido [Xu et al., 1997]. Qualquer processo (emissor ou um dos receptores) que receber este NACK e tiver o quadro solicitado, avalia novamente os parâmetros (linhas 33 e 36) e, se for o caso, difunde-o

novamente no grupo (linhas 38). Note que esta avaliação de parâmetros de QoS para retransmissão seletiva é feita apenas para pacotes P e B. Pacotes do tipo I perdidos são sempre (tentativamente) recuperados (linhas 13-15 e 30-32).

Algoritmo do Emissor
1. multicast(m) <i>{envia a mensagem de dados diretamente na rede}</i>
Algoritmo do Receptor
2. QUANDO recebe(m)
3. SE m.tipo = DADOS <i>{se recebeu uma mensagem de dados}</i>
4. quadro = m.tipo_do_quadro
5. SE busca_no_buffer(m.emissor, m) == NULL
6. cancela_agendado(NACK _m) <i>{cancela envio de NACK para m}</i>
7. adiciona_ao_buffer(m.emissor, m)
8. entrega(m) <i>{entrega a mensagem}</i>
9. SENÃO
10. cancela_agendado(m) <i>{cancela retransmissão de m}</i>
11. FIM-SE
12. SE há mensagem faltando <i>{recuperação de mensagens faltando}</i>
13. SE quadro == "I"
14. espera(aleatório(T _{nack}))
15. multicast(NACK _m)
16. SENÃO SE ainda_relevante(quadro)
17. taxa_de_perda = verifica_taxa_de_perda()
18. SE (quadro == "P" E taxa_de_perda < 40%) OU
19. (quadro == "B" E taxa_de_perda < 20%)
20. espera(aleatório(T _{nack}))
21. multicast(NACK _m)
22. FIM-SE
23. FIM-SE
24. FIM-SE
25. SENÃO SE m.tipo == NACK <i>{se recebeu um pedido de retransmissão}</i>
26. PARA TODOS m _n ∈ m.nacked
27. cancela_agendado(NACK _{m_n}) <i>{cancela envio de NACK para m_n}</i>
28. m _T = busca_no_buffer(m.emissor, m _n)
29. SE m _T ≠ NULL
30. SE m _T .quadro == "I"
31. espera(aleatório(T _{reparo}))
32. multicast(m _T)
33. SENÃO
34. taxa_de_perda = verifica_taxa_de_perda()
35. SE (quadro == "P" E taxa_de_perda < 40%) OU
36. (quadro == "B" E taxa_de_perda < 20%)
37. espera(aleatório(T _{reparo}))
38. multicast(m _T)
39. FIM-SE
40. FIM-SE
41. FIM-SE
42. FIM-PARA
43. FIM-SE
44. FIM-QUANDO

4.2. Retransmissões

O protocolo baseia-se no princípio básico de retransmissão efetuada pelos receptores, onde estes compartilham a responsabilidade de ajudar os demais receptores a recuperarem as perdas [Floyd et al., 1995]. Para análise dos parâmetros de decisão, convencionou-se que a taxa de perda máxima para pedir retransmissão ou retransmitir pacotes contendo quadros do tipo B é de 20% e 40% para pacotes contendo quadros do tipo P (linhas 18-19 e 35-36). Estes valores foram determinados experimentalmente.

A taxa de perda em cada receptor é calculada a partir de uma seqüência de amostragem, os últimos N pacotes difundidos. A cada N pacotes dessa seqüência, verifica-

se a quantidade de pacotes recebidos detectando possíveis lacunas na sequência. Portanto, a função *verifica_taxa_de_perda* (chamada nas linhas 17 e 34) calcula a porcentagem de pacotes perdidos a cada N pacotes difundidos.

A relevância de um quadro é verificada através do *buffer* de reprodução. A função *ainda_relevante* (chamada na linha 16) checa se o quadro que está faltando já está obsoleto, ou seja, se o momento de sua reprodução pela aplicação já passou.

Para evitar tanto a explosão de NACKs como a de retransmissões, é usada uma função de espera que interrompe a execução durante um tempo aleatório (cujos limites superiores são T_{nack} e T_{reparo}). Quando um receptor R_i está prestes a enviar um NACK, ele espera por um tempo aleatório T_{nack} (linhas 14 e 20). Se durante este tempo R_i receber um NACK de um outro receptor R_j , pedindo pelos mesmos pacotes, R_i cancela a difusão do seu NACK (linha 27). De modo similar, quando um receptor R_i recebe um NACK e pode atendê-lo, R_i espera por um tempo aleatório T_{reparo} antes de retransmitir o(s) pacote(s) solicitado(s) nesse NACK (linhas 32 e 38). No entanto, se durante o tempo T_{reparo} R_i receber alguns ou todos os pacotes solicitados no NACK, R_i pode cancelar a difusão dos pacotes já retransmitidos (linha 10).

4.3 Buffer de Reprodução e Buffer de Retransmissão

Um ponto importante para a implementação do protocolo consiste no gerenciamento do *buffer* de reprodução no receptor de forma a tratar adequadamente as perdas indiretas, que ocorrem devido à perda de um quadro do tipo I. Para solucionar o problema, foram criadas duas classes: *WaitElement* e *WaitWindow*. A classe *WaitElement* representa um conjunto de pacotes recebidos onde o primeiro deles contém um quadro do tipo I. Um objeto da classe *WaitElement* tem três campos: o número de sequência do quadro I daquele conjunto, um *flag* que indica se o quadro I já está presente no *buffer* auxiliar e o próprio *buffer* auxiliar, que armazena os pacotes daquele conjunto. A figura 2 ilustra este objeto.

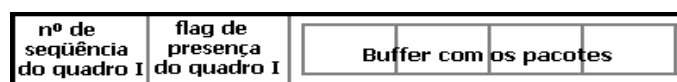


Figura 2. Classe *WaitElement*

A classe *WaitWindow* representa um *buffer* de armazenamento dos objetos *WaitElement*. Os pacotes que chegam ao receptor são colocados dentro do *WaitElement* correspondente, de acordo com o número de sequência do quadro que contém. Os objetos *WaitElement* são armazenados no objeto *WaitWindow* do receptor, ordenados pelo número de sequência do quadro I do conjunto. A figura seguinte mostra como ficaria instanciada a classe *WaitWindow*, para a uma dada sequência de quadros.

Para cada pacote recebido contendo um quadro I que chega, é criada uma nova instância da classe *WaitElement* que é adicionada ao objeto *WaitWindow* na posição correspondente, de acordo com o número de sequência do quadro I. Para os pacotes contendo os demais tipos de quadro, procura-se dentro do objeto *WaitWindow*, o *WaitElement* que contenha o quadro I que o precede, e o pacote é adicionado ao respectivo *buffer* na posição correta.

Se for detectada a perda de um pacote contendo um quadro do tipo I, é criada uma instância de *WaitElement* com o número de sequência do pacote que foi perdido e com o *flag* assumindo o valor falso, indicando que o quadro I daquele conjunto não está no *buffer*. Dessa forma, se chegar o momento de reproduzir este conjunto de quadros antes da recepção do respectivo quadro I, todos os quadros daquele conjunto serão descartados para

apresentação e o próximo conjunto que contenha um quadro I é que será colocado no *buffer* de reprodução.

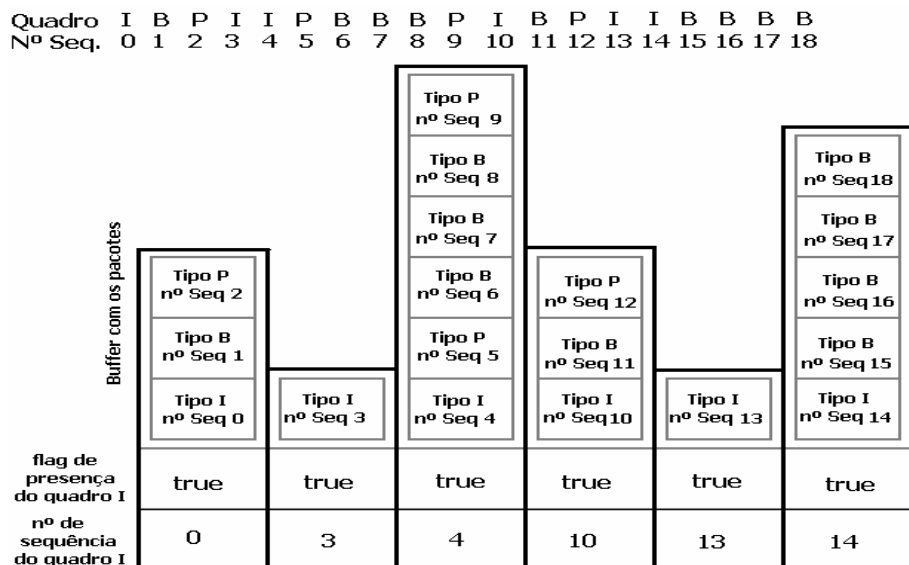


Figura 3. Objeto da Classe WaitWindow

Esse algoritmo permite que, quando os quadros são enviados para o *buffer* de reprodução, eles já estejam ordenados e que os descartes de quadros perdidos já tenham sido realizados. A retransmissão também ocorre a partir do objeto *WaitWindow*, uma vez que ele armazena os pacotes até o momento da reprodução dos quadros.

Empregando esta estrutura, cada receptor deve possuir uma entidade para cuidar do envio dos quadros para a aplicação. Isto deve ser efetuado passando os quadros que estão na *WaitWindow* para o *buffer* de reprodução que será lido pela aplicação.

Cada receptor possui um *buffer* de retransmissão que guarda os últimos pacotes (quadros) recebidos. Os quadros que estão no *buffer* de reprodução são os mesmos que estão no *buffer* de retransmissão sem a necessidade de redundância. O *buffer* de retransmissão é utilizado por um receptor para atender pedidos de retransmissão de pacotes (NACK). Este *buffer* tem tamanho limitado e os pacotes mais antigos são descartados quando seu limite é atingido utilizando-se o método FIFO (*First In First Out*).

5. Simulações e Resultados

Para avaliar o protocolo proposto, utilizou-se o simulador de rede Simmcast [Barcellos, 2001]. Foram feitas comparações de desempenho em termos de recuperação de erros, tempo de recuperação, sobrecarga nos receptores e fator de qualidade de vídeo [Martins, 2003] entre o protocolo proposto, um protocolo de *multicast* simples e um protocolo de *multicast* com envio de NACKs e retransmissões a partir da fonte. A topologia adotada nas simulações está ilustrada na figura 4 e visa mostrar a potencialidade do protocolo proposto em redes de larga escala.

As simulações foram realizadas para os três protocolos utilizando-se as mesmas condições de envio, ou seja, com os mesmos dados sendo transmitidos sob as mesmas condições de rede. Os parâmetros usados no protocolo proposto são (seção 4.2): $N = 290$, $T_{reparo} = 200$ e $T_{nack} = 200$ (ambos em ms).

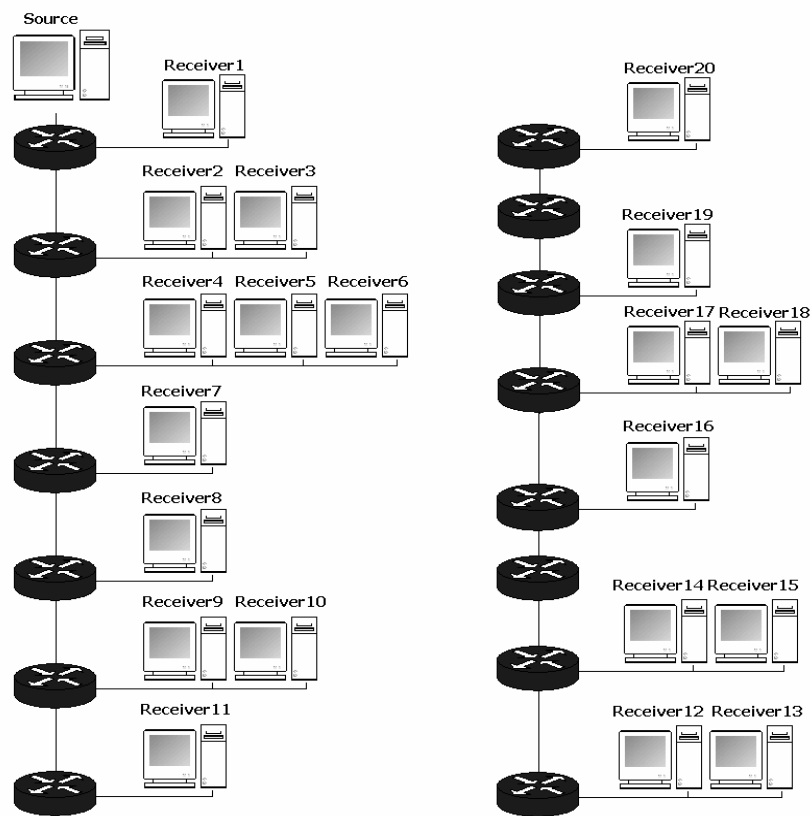


Figura 4. Topologia utilizada nas simulações

Em termos de recuperação de pacotes não recebidos, o protocolo proposto apresentou uma média de 43% de recuperação (de todos os quadros perdidos) contra 11% do *multicast* com retransmissão simples. Quando analisados somente os pacotes que contêm quadros do tipo I, o protocolo proposto apresenta uma média de 41,6% de recuperação contra 0,6% do *multicast* com retransmissão simples. Isso mostra que as retransmissões seletivas aumentam a eficácia na recuperação, especialmente dos quadros mais importantes. Os valores médios de recuperação por tipo de quadro são mostrados na Tabela 1.

	% de quadros recuperados		
	I	P	B
Protocolo Proposto	41.6	68.35	37.0
<i>Multicast</i> com envio de NACK	0.6	3.15	1.7

Tabela 1. Recuperação por tipo de quadro

Outro aspecto interessante a ser observado é que a média de pacotes recuperados varia menos entre os receptores quando usado o protocolo proposto. Para o *multicast* com retransmissão simples, esta média diminui bruscamente à medida que o receptor está mais distante da fonte transmissora, como ilustra o gráfico 1.

Observou-se que, para o protocolo proposto, a grande maioria dos pacotes recuperados nos receptores (89.03%) vêm de seus vizinhos e não da fonte. Isto implica em um desempenho melhorado em termos de tempo de recuperação de pacotes perdidos. Para a topologia utilizada, o protocolo proposto apresenta uma média de 104 ms de tempo de recuperação contra 170 ms do *multicast* com retransmissão.

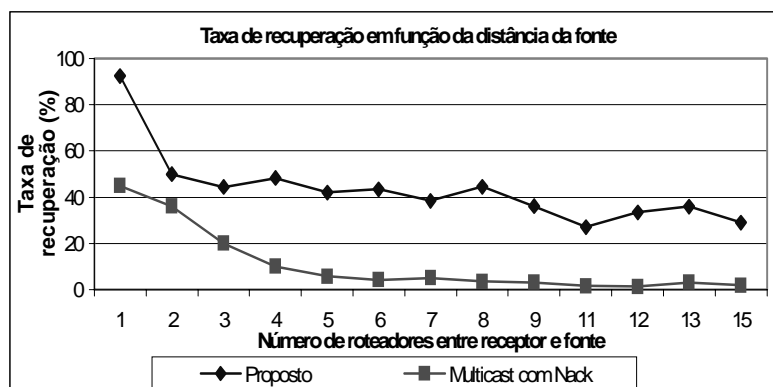


Gráfico 1. Recuperação em função da distância da fonte

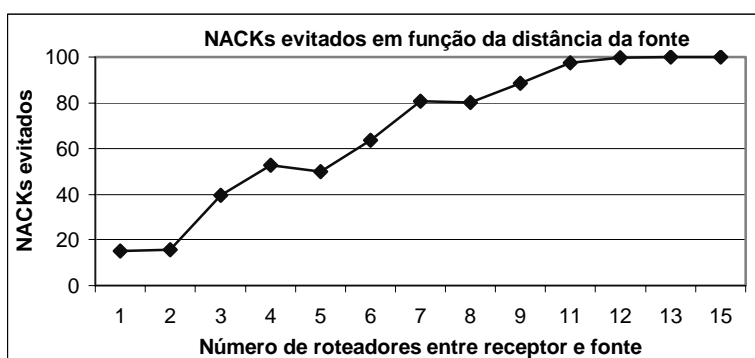


Gráfico 2. NACKs evitados em função da distância da fonte

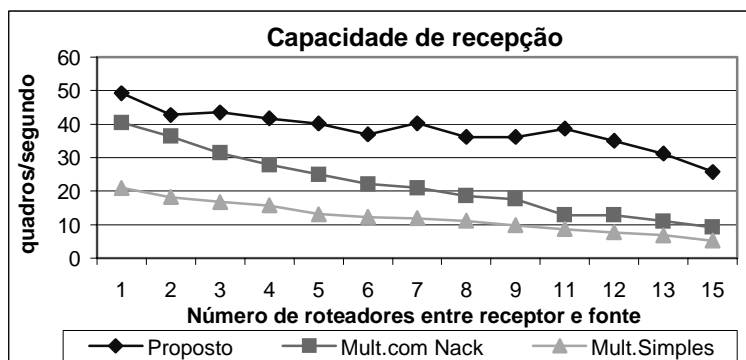


Gráfico 3. Capacidade de recepção.

Quanto à sobrecarga causada na rede, o protocolo proposto apresentou um aumento de aproximadamente 55% na largura de banda utilizada para efetuar as retransmissões em relação ao *multicast* sem retransmissão. Este aumento é quase quatro vezes maior do que o *multicast* com retransmissão simples, que aumenta apenas 14,4% a largura de banda consumida. Nota-se que este aumento é proporcional à melhoria obtida na taxa de quadros recuperados (43% para 11%, conforme mostrado anteriormente), sendo um resultado de certa forma esperado uma vez que, no protocolo proposto, as retransmissões ocorrem a partir de todos os nós da rede e não somente da fonte. Um resultado bastante positivo foi a omissão de NACKs no protocolo proposto, em média 39,2% dos NACKs foram evitados. Observou-se que essa porcentagem aumenta em função da distância da fonte, como mostra o Gráfico 2, chegando a 100% nos receptores mais afastados (*Receiver19* e *Receiver20* na Figura 4). Isto ocorre porque os receptores mais próximos da fonte detectam as lacunas na

seqüência mais rapidamente enviando seus NACKs antes que os receptores mais afastados possam fazê-lo.

A capacidade de recepção no protocolo proposto é ilustrada no Gráfico 3. Esta capacidade varia em função da distância da fonte no cenário da Figura 4. É possível perceber que o protocolo proposto mostra-se muito mais adequado para a transmissão de mídia contínua do que os demais uma vez que, mesmo para os nós mais afastados, a taxa de recepção fica acima dos 25 quadros/segundo necessários neste tipo de transmissão.

Um outro parâmetro adotado para avaliação do protocolo foi o *Fator de Qualidade de Vídeo* (q), definido em [Martins et al., 2003]. Este parâmetro consiste em uma métrica baseada na estrutura do GOP para avaliar a qualidade de vídeo. Sua fórmula, representada pelas equações 1 e 2, leva em conta tanto as perdas diretas, ocasionadas pela perda do próprio quadro, quanto as indiretas, causadas pela falta de um outro quadro (por exemplo, um quadro P ou B que não pode ser reproduzido pela falta de um quadro I). É importante lembrar que essa métrica busca avaliar as informações do fluxo de vídeo que são transportadas, mas não avalia diretamente a qualidade do ponto de vista do usuário final.

$$q = \frac{a_I * x_I + a_P * x_P + a_B * x_B}{a_I * N_{TI} + a_P * N_{TP} + a_B * N_{TB}}, 0 \leq q \leq 1 \quad (1)$$

onde,

J : Representa o tipo de quadro (se I, P ou B)

x_i : Representa o número de quadros do tipo j que são recebidos e mostrados

N_{Tj} : Representa o número total de quadros do tipo j presentes em um GOP

a_j : Representa o coeficiente relativo (quadros do tipo j no GOP)

e,

$$a_j = \frac{N_{Ij}}{N_{II} * N_{TI} + N_{IP} * N_{TP} + N_{IB} * N_{TB}} \quad (2)$$

onde,

N_{Ij} : Representa as perdas diretas e indiretas provocadas pela perda do quadro j

Verificou-se nos três protocolos testados uma variação considerável em função da distância. Como pode ser observado no Gráfico 4, quanto mais próximos os receptores, melhor o valor de q . No entanto, para o protocolo proposto a tendência é de que a queda do fator q em função da distância da fonte seja mais lenta que nos demais protocolos.

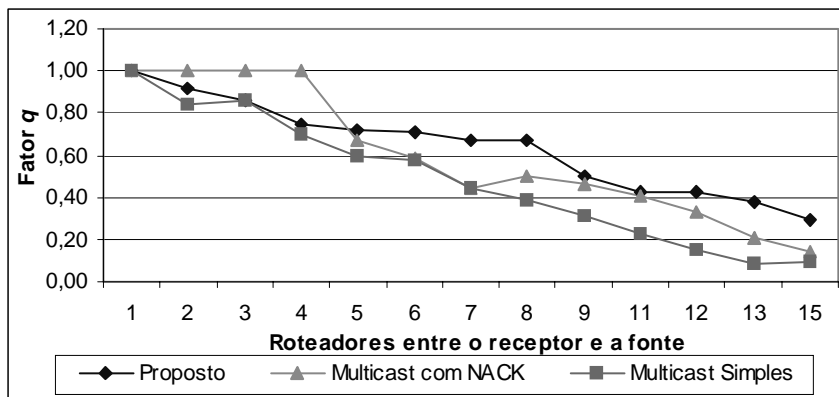


Gráfico 4. Fator q em função da distância da fonte

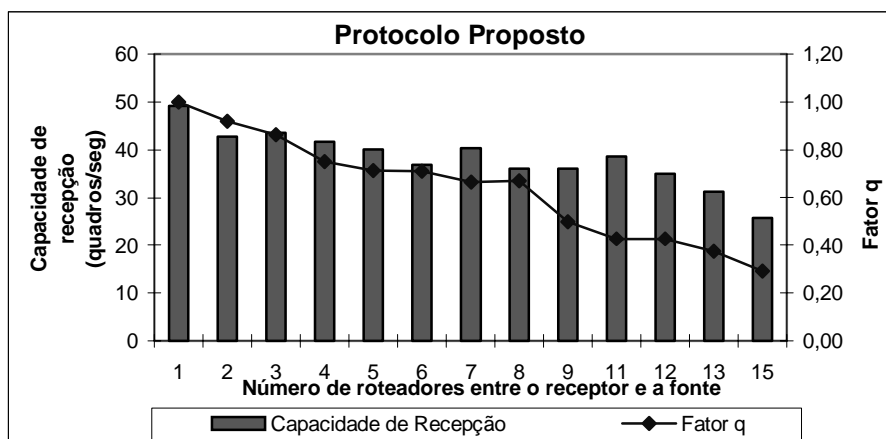


Gráfico 5. Análise geral do protocolo proposto

O Gráfico 5 mostra que existe uma relação direta entre a capacidade de recepção e a variação do fator q .

6. Trabalhos Relacionados

Na literatura, são encontrados alguns protocolos para *multicast* semi-confiável. Dentre eles, pode-se destacar o **WAIT [Mane et al., 2000]**, projetado para ser ajustado para diferentes requisitos de qualidade de uma sessão multimídia e reduzir a carga na rede, além de uma melhor qualidade de serviço nas aplicações sobre a Internet com menor suporte do roteador. O problema desta abordagem é que, na formação de um grupo, os receptores precisam ter alguma informação sobre a topologia em que se encontram e isso reduz a escalabilidade do protocolo.

Pode-se citar também o **SRP [Piecuch, 2000]** (*Selective Retransmission Protocol*), que usa um algoritmo de decisão específico por aplicação para determinar se um pedido de retransmissão do pacote perdido deve ser enviado ou não, ajustando o nível de perda e latência de acordo com a aplicação. Apenas uma parte dos dados perdidos é retransmitida, sendo que a quantidade depende dos fatores de QoS incluindo perdas atuais e latência, RTT (*round trip time*), congestionamento da rede e qualidade requisitada pelo usuário.

Destaca-se ainda o **TUNA (TUNable Quasi-Reliable Multicast Protocol) [Wong et al., 1998]**, uma adaptação da confiabilidade seletiva para disseminação de informações periódicas. A principal idéia do TUNA é suprimir NACKs de um receptor quando a “idade” esperada da informação perdida está abaixo de um limite especificado pelo usuário. Em outras palavras, se a idade for maior que o limite, o receptor imediatamente envia um NACK. Senão, ele simplesmente espera pela próxima atualização da informação perdida. Assim, os receptores não têm garantia de receber tudo do emissor (por isso semi-confiável) mas a idade da informação é mantida dentro de um limite (é ajustável).

Uma outra abordagem interessante é apresentada por Pereira et al. em seus trabalhos sobre protocolos de difusão semanticamente confiáveis **[Pereira et al., 2003]**. É proposto um modelo de protocolo de *multicast* confiável que elimina mensagens obsoletas para sustentar um *throughput* mais alto. A idéia é de que a obsolescência da mensagem só seja utilizada para evitar congestionamentos. Quando a ocupação do *buffer* sobe além do limite, o protocolo busca mensagens obsoletas neste *buffer* e as elimina. Assim que a ocupação do *buffer* baixa, o protocolo retoma a operação confiável. Tanto o emissor quanto o receptor podem eliminar mensagens obsoletas encontradas em seus *buffers* locais.

Os resultados das implementações comprovaram que protocolos de *multicast* semanticamente confiáveis resultam em uma melhoria da estabilidade do *throughput* mesmo com receptores com desempenho mais limitado.

O **PRTP (Partially Reliable Transport Protocol)** [Schneyer,1999] apresenta um serviço parcialmente confiável que não insiste em recuperar todos erros, mas apenas uma parte dos dados perdidos. Ele melhora a entrega dos pacotes permitindo que as aplicações troquem uma quantidade controlada de perdas por um maior *throughput*, ou seja, as próprias aplicações definem a quantidade mínima de confiabilidade. Quando o parâmetro *nível de confiabilidade* está acima do limite, o receptor não pede retransmissão e envia um ACK. Se um pacote é perdido, o receptor verifica se o nível de confiabilidade está acima do limite. Se estiver, o receptor envia um ACK. O processo é repetido tantas vezes quanto necessário enquanto não violar o limite estabelecido pela aplicação. Este protocolo pode ser considerado benéfico quando estão correndo muitas perdas no canal e os RTTs não são insignificantes, como o caso de ambientes sem fio.

Yavatkar e Manoj propuseram um protocolo de transporte *multicast* que ele chamou quasi-confiável para transmissão de informação multimídia em larga escala [Yavatkar e Manoj,1993]. Afirma-se que, dada a natureza da comunicação de multimídia, o protocolo deve usar a técnica FEC (*Forward Error Correction*) para evitar os atrasos inerentes às técnicas baseadas em fluxo e controle de erro.

O protocolo STORM (*STructure-Oriented Resilient Multicast*) [Xu et al.,1997] é baseado em uma estrutura de distribuição. Os participantes do grupo organizam-se em uma estrutura de distribuição que vai ser usada na recuperação de pacotes dos nós adjacentes. Além disso, esta estrutura é dinâmica e um algoritmo é usado para adaptá-la às condições de tráfego da rede e às condições dos membros do grupo.

O **SRM (Scalable Reliable Multicast)** [Floyd, 1995] um protocolo de difusão confiável e serviu de base para o sistema de retransmissão do protocolo proposto. Neste protocolo sempre que é detectada uma perda, um NACK é enviado para todo o grupo e qualquer membro que possua o pacote perdido pode retransmiti-lo. Para evitar NACKs ou pacotes duplicados, o nó estabelece um tempo aleatório antes de enviá-los. Se o nó receber o NACK ou pacote que iria enviar, cancela este envio.

O **PRMP (Polling-based Reliable Multicast Protocol)** [Barcellos, 98] é um protocolo de *multicast* confiável baseado no emissor que busca resolver o problema de escalabilidade limitada através de um mecanismo para evitar a implosão baseada em eleição. Os receptores são escolhidos em momentos cuidadosamente planejados de forma que, apesar dos diferentes conjuntos de RTTs, a taxa de pacotes *de feedback* que cheguem à fonte não excedam a capacidade do emissor ou da rede.

A maior inovação do protocolo proposto em relação aos protocolos de *multicast* semi-confiável encontrados na literatura é a utilização da hierarquia dos quadros MPEG para tomada de decisão sobre as retransmissões.

7. Conclusão

Este artigo apresentou um protocolo para otimização da utilização dos dados de vídeo que trafegam pela rede. Com base no padrão MPEG e na tecnologia *multicast*, este protocolo garante a entrega confiável dos quadros de vídeo, ou de parte deles. Seu desempenho é obtido pelo método de retransmissão baseado no receptor que utiliza o descarte seletivo dos pacotes perdidos de acordo com a importância do quadro MPEG neles contido.

Os resultados das simulações foram bastante satisfatórios, especialmente o tempo médio de recuperação de pacotes e a porcentagem de quadros recuperados. No momento,

estamos investigando formas de determinar os parâmetros que limitam os envios de NACK e pedidos de retransmissão dos pacotes contendo quadros do tipo P e B utilizando técnicas adaptativas para que o protocolo atinja um desempenho ótimo. Neste contexto, nos parece interessante também investigar algoritmos de controle de fluxo e congestionamento [Barcellos e Roesler, 2000] aplicados em protocolos de *multicast* semi-confiáveis.

Em comparação com o *multicast* simples e o multicast com envio de NACK, percebe-se que o protocolo proposto é mais eficiente para redes de larga escala, apresentando resultados melhores nos receptores mais afastados da fonte.

Referências

- Barcellos, Marinho P. (1998). "PRMP: A Scaleable Polling-based Reliable Multicast protocol". Ph.D. Thesis, Department of Computing Science, Newcastle University, Newcastle upon Tyne, October 1998, 200p.
- Marinho P. Barcellos, Valter Roesler (2000), "M&M: Multicast e Multimídia", SBC/JAI2000 - XX Congresso da Sociedade Brasileira de Computação - pp.203-244. Julho de 2000.
- Barcellos, Marinho et al. (2001). "Simmcaster: a Simulation Tool for Multicast Protocol Evaluation". Em: XI Simpósio Brasileiro de Redes de Computadores, 2001, Florianópolis.
- Bessani, Alysson N., Lung, Lau C., Fraga, Joni da S. (2003). "ReMIOP: Projeto e Implementação de um Mecanismo de Difusão Confiável no CORBA" Em: XXI Simpósio Brasileiro de Redes de Computadores, 2003, Natal. SBRC'03. Porto Alegre: SBC, 2003. p. 957-964.
- Chiariglione, Leonardo (2000). "Short MPEG-2 Description", Abril de 2000. Disponível em <http://mpeg.telecomitalialab.com/standards/mpeg-2/mpeg-2.htm>.
- Floyd, S. et al (1995) "A reliable multicast framework for light-weight sessions and application level framing". Em: Proceedings of the ACM SIGCOMM 95, Agosto de 1995, p.342-356
- Hadzilacos, V. and Toueg, S. (1994). "A modular approach to the specification and implementation of fault-tolerant broadcasts". Technical report, Department of Computer Science, Cornell University, New York - USA.
- Liao, T (1998). "Light-weight reliable multicast protocol". Disponível em <http://webcanal.inria.fr/lrmp/>.
- Mane, Pravin. (2000) "WAIT: Selective Loss Recovery For Multimedia Multicast", MS Thesis - Computer Science Department, WPI. Maio de 2000.
- Martins, R. F. Leite, C. A., Farines, J-M (2003), "Toward Quality Evaluation and Improvement of a MPEG Vídeo Stream". Proc. Of the 3th IEEE Latin American Network Operations and Management Symposium – LANOMS'03, Foz do Iguassu, Brasil, 2003.
- Pereira, Jose et al (2003) "Semantically Reliable Multicast: Definition, Implementation, and Performance Evaluation". IEEE Transactions on Computers, vol.52, nº2, p150-165, 2003.
- Piecuch, Mike et al. (2000) "A Selective Retransmission Protocol for Multimedia on the Internet". Em: Proceedings of SPIE International Symposium on Multimedia Systems and Applications. Novembro de 2000
- Schneyer, S., Garcia, J., Brunstrom, A., Asplund, K. (1999) PRTP: "A Partially Reliable Transport Protocol for Multimedia Applications", Proceedings Int. Symposium on Intelligent Multimedia and Distance Education (ISIMADE), Baden-Baden, Germany, August 1999.
- Wong, Tina et. al. (1998) "Exploiting Application Level Framing in Reliable Multicast for Periodic Information Dissemination". Poster in the Proceedings of ACM Multimedia '98, Bristol, UK, Setembro de 1998.
- Yavatkar, Rajendra e Manoj, Leelanivas (1993) "Optimistic Strategies for Large-Scale Dissemination of Multimedia Information", In: Proceedings of the First ACM International Conference on Multimedia '93.
- Xu, X., Myers, A., Zhang, H. and Yavatkar, R. (1997), "Resilient Multicast Support for Continuous-Media Applications," in Intl. Workshop on Network and OS Support for Digital Audio and Video (NOSSDAV97).