Comunicação Multiponto Confiável em Redes Ativas Utilizando Cache Adaptativo

Flavio Paiva Junqueira
flavio@gta.ufrj.br

Otto Carlos M. B. Duarte *
otto@gta.ufrj.br

Grupo de Teleinformática e Automação - GTA
COPPE/EE - Programa de Engenharia Elétrica
Universidade Federal do Rio de Janeiro
C. P. 68504 - CEP 21945-970 - Rio de Janeiro - Brasil

*Este trabalho foi realizado com apoio financeiro de: UFRJ, FUJB, REENGE, CNPq, CAPES e COFECUB.

Título: Comunicação Multiponto Confiável em Redes Ativas Utilizando Cache Adaptativo

Resumo

Este artigo propõe um serviço de comunicação multiponto confiável baseado em nós ativos, que executam procedimentos específicos deste serviço com os objetivos de filtrar o excesso de cápsulas, para evitar o problema de implosão, e realizar retransmissões localizadas através do uso de *cache* de dados, para evitar o desperdício de banda passante e agilizar a recuperação de dados. A alocação de memória do *cache* é realizada por um mecanismo adaptativo capaz de manter este tamanho em um valor mínimo quando a taxa de perdas é baixa e aumentá-lo de acordo com a incidência de pedidos de retransmissão. Os resultados obtidos por simulações demonstram o bom desempenho do serviço utilizando a alocação adaptativa, mesmo com altas taxas de erro.

Abstract

This paper proposes a reliable multicast communication service based on active nodes that execute specific procedures of this service to accomplish capsule filtering, to avoid the implosion problem, and scoped retransmissions through data caches, to save bandwidth and to carry out faster recoveries. We propose an adaptive mechanism to allocate cache space that is capable of maintaining a minimum size when loss rates are low, and increasing it when the number of repair requests augment. Simulation results show that this service performs well, even for multicast receivers experiencing high loss rates.

1 Introdução

Nos últimos anos, a Internet tem passado por um processo de expansão, caracterizado pelo aumento do número de usuários e pela mudança no perfil dos mesmos. Com isso, surgiu uma demanda por novas aplicações que necessitam de serviços de rede com características diferentes daquelas apresentadas pelo serviço de melhor esforço oferecido atualmente na Internet. Entre estas aplicações encontram-se aquelas em que participam vários usuários, denominadas aplicações multipontos, que possuem diferentes requisitos, principalmente com relação à confiabilidade [1]. Aplicações multipontos que suportam interatividade, tais como videoconferência e apresentações multimídias envolvendo áudio e/ou vídeo, não requerem transferência de dados confiável, enquanto que aplicações tais como simulação interativa distribuída e *whiteboard* compartilhado apresentam esta necessidade [2, 3].

A confiabilidade na transferência de dados de aplicações multipontos gera uma problema conhecido como implosão de reconhecimentos [1, 2, 4]. Quando um número grande de usuários é desejável para um grupo multiponto, o envio de múltiplos reconhecimentos de mensagens de dados pode ocasionar congestionamentos e sobrecarga no transmissor, devido à limitações de banda passante e capacidade de processamento, respectivamente. Além disto, os receptores, em geral, enfrentam diferentes taxas de perda dependendo da sua localização na árvore multiponto [4]. Sendo assim, a retransmissão de dados para todo o grupo desperdiça banda passante e afeta o desempenho da aplicação.

Os problemas apresentados para as aplicações multipontos confiáveis podem ser resolvidos através da execução de computações e armazenamento nos nós intermediários para evitar a implosão de reconhecimentos e realizar retransmissões localizadas. Esta proposta segue o paradigma das pesquisas correntes em redes ativas, que tem como principal objetivo a inserção de computações específicas das aplicações nos nós intermediários da rede, denominados nós ativos [5]. Computações realizadas no interior da rede poderiam melhorar o desempenho das aplicações através da diminuição da quantidade de dados trafegando, pois as execuções não estariam mais restritas às entidades finais [6, 7, 8].

Algumas abordagens para redes ativas propõem que os novos serviços disponíveis para as aplicações sejam construídos a partir da implementação de procedimentos que seriam carregados nos pacotes e executados nos nós ativos. Tennenhouse *et al.* [6, 9] do MIT propõem o uso de pacotes, denominados de cápsulas, que carregam além dos dados, uma referência para um procedimento a ser executado a cada nó ativo, sendo que o carregamento dos procedimentos é feito dinamicamente. Nesta proposta, diversos recursos são disponibilizados pelos nós para o uso dos procedimentos, entre eles o armazenamento não-transiente de estados, mais conhecido como *soft-state*. Utilizando este recurso, as cápulas podem armazenar quaisquer informações necessárias para a decisão do destino das próximas cápsulas do seu fluxo. Neste contexto, fluxo é uma seqüência de cápsulas que seguem o mesmo caminho e requerem o mesmo processamento.

Tendo em vista as vantagens das execuções realizadas no interior da rede, Tennenhouse *e*t al. [4]propuseram ainda um esquema escalável, denominado ARM (*Active Reliable Multicast*), para a realização de retransmissões localizadas em comunicações multiponto na Internet, utilizando *caches* de dados nos nós intermediários. Este esquema se utiliza de um protocolo multiponto orientado ao receptor, em que reconhecimentos negativos (NAKs) são enviados pelos receptores do grupo multiponto quando saltos são detectados no seqüenciamento dos pacotes recebidos. Além disto, para proteger o transmissor de pedidos de retransmissão simultâneos, o ARM propõe o uso de filtros nos nós intermediários para descartar pedidos repetidos. A principal vantagem deste esquema se encontra na robustez com relação à falhas na rede, pois tanto o armazenamento em *cache* quanto a filtragem de pedidos de retransmissão excessivos são realizadas nos nós que formam a árvore multiponto, que pode ser facilmente rearranjada.

No entanto, o *cache* utilizado no ARM baseia-se em uma política de melhor esforço, isto é, a quantidade de pacotes que podem ser armazenados depende do espaço de armazenamento existente em um nó e que não está sendo utilizado para outros fins. Esta política nem sempre é conveniente, pois a garantia de um espaço mínimo para o armazenamento de dados multipontos algumas vezes é essencial para um bom desempenho do serviço, como por exemplo para o serviço de comunicação multiponto para estações móveis proposto em [10].

Este trabalho propõe um serviço para comunicação multiponto confiável que se utiliza de um protocolo orientado ao receptor e de um mecanismo de alocação de *cache* adaptativo, isto é, o tamanho do *cache* de dados para um grupo multiponto pode variar durante o tempo de existência do grupo multiponto. A variação do tamanho do *cache* ocorre de acordo com as condições da rede, ou seja, o tamanho do *cache* permanece mínimo se não houverem pedidos de retransmissão e aumenta conforme a incidência destes pedidos. Este mecanismo de alocação tem como principais vantagens a economia de recursos quando houverem baixas taxas de erro e o benefício de um maior índice de retransmissões localizadas quando as taxas de perda forem maiores. Simulações foram realizadas para este serviço com o objetivo de verificar o seu desempenho e a quantidade de recursos de memória utilizada.

O restante deste artigo está dividido como se segue. A seção 2 apresenta o serviço proposto e o protocolo utilizado para este serviço. A configuração dos parâmetros do *cache* são descritos na seção 3. O modelo de simulação e os resultados são apresentados na seção 4. Por último, a seção 5 apresenta as conclusões deste trabalho.

2 O Serviço Proposto

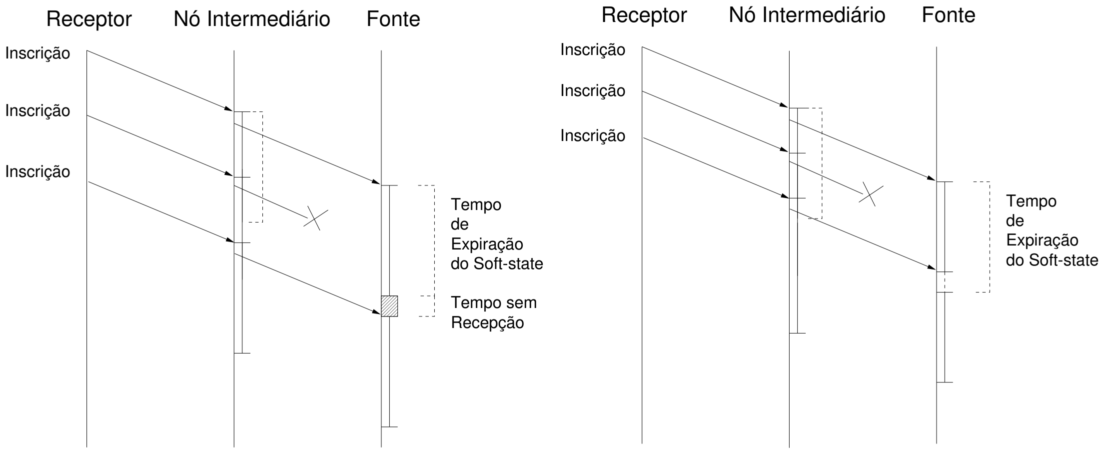
O serviço utiliza o mecanismo proposto por Tennenhouse *et al.* para a formação e manutenção de árvores ponto-a-multiponto através de cápsulas de inscrição enviadas pelos receptores e o uso do *soft-state* nos nós ativos para guardar o caminho do transmissor aos receptores. A escolha deste mecanismo deve-se ao seu dinamismo e tolerância a falhas na rede, uma vez que a reconfiguração dos caminhos pode ser facilmente realizada. A cápsula de inscrição deve percorrer o caminho até o transmissor, verificando nos nós ativos a existência de marcações referentes ao grupo multiponto de interesse. Ao encontrar, a adesão ao grupo está completa e este receptor passa a receber as mensagens multipontos, mesmo que a cápsula de inscrição precise percorrer o restante do caminho até o transmissor para atualizar o tempo de expiração do *soft-state* dos próximos nós ativos.

As cápsulas de dados multipontos são enviadas para os receptores seguindo o caminho deixado pelas cápsulas de inscrição. Assim, o destino das cápsulas de dados multipontos não é determinado pelo endereço da máquina e sim pelo caminho marcado no *soft-state*.

A confiabilidade do serviço é conseguida através de retransmissões realizadas pelo transmissor e pelos nós ativos intermediários, como é proposto no ARM. *Caches* de cápsulas de dados multipontos devem existir nos nós ativos intermediários para a recuperação mais rápida de perdas e dentro do escopo das perdas. O tamanho do *cache* é negociado pelas cápsulas de inscrição, calculando os valores mínimo e máximo segundo as equações apresentadas na seção 3.

O protocolo proposto para suportar este serviço é composto de quatro cápsulas: cápsula de inscrição, cápsula de dados multiponto, cápsula de pedido de retransmissão e cápsula de fim de sessão.

As cápsulas de inscrição são enviadas periodicamente para o transmissor de um grupo multiponto por todas as estações interessadas em receber os dados deste grupo. O período de envio de cápsulas de inscrição depende do tempo de expiração do *soft-state* nos nós ativos. Estas cápsulas devem ser enviadas dentro de um período de tempo que permita manter a continuidade da comunicação, mas mantendo o compromisso de não inundar a rede com cápsulas de inscrição. Assim, o período de envio (T_e) deve ser menor que o período de expiração do *soft-state* (T_s) para compensar atrasos em filas. A figura 3 ilustra duas situações onde $T_e < T_s$. Para o caso 1, tem-se que $T_s > T_e > 0.5 \times T_s$. Neste caso, se a cápsula de inscrição for perdida no seu percurso, haverá perda de continuidade da comunicação. Para o caso 2, tem-se que $0.5 \times T_s > T_e$. Neste caso, a ocorrência de uma perda não acarreta em descontinuidade da comunicação, pois a segunda cápsula de inscrição enviada percorreu o caminho a tempo de atualizar o *soft-state*. Só haveria descontinuidade na comunicação para a situação 2 em caso de congestionamento no caminho do receptor para a fonte. O valor percentual de T_s utilizado como período de envio pode ser calculado supondo um número maior de perdas de cápsulas de inscrição. No entanto, o tempo extra conferido para permitir a perda de uma cápsula confere uma alta robustez ao mecanismo, principalmente considerando-se que outros usuários que compõem o grupo também estão enviando cápsulas de inscrição.



(a) Caso 1

(b) Caso 2

Figura 1: Envio de inscrição.

O procedimento de inscrição utiliza o valor de T_e para filtrar as cápsulas de inscrição em excesso, ou seja, somente as inscrições necessárias para manter os *soft-states* dos nós ativos atualizados devem passar. Isto é verificado através da estampa de tempo colocada pelo nó ativo a cada vez que o *soft-state* é atualizado. Além desta estampa para o controle do *sof-state* como um todo, cada entrada no *soft-state* do grupo deve conter uma estampa de tempo. Uma das funções do procedimento de inscrição é verificar as entradas expiradas e removê-las. Se isto não fosse feito no *soft-state* de um grupo em que existem várias indicações, não seria possível a remoção de ramos que não pertencem mais à árvore. Neste caso, cápsulas de dados multipontos seriam enviadas desnecessariamente.

As cápsulas de dados multipontos seguem as marcações deixadas pelas cápsulas de inscrição até atingirem os receptores. Assim, a função principal destas cápsulas é ler as indicações contidas no *soft-state* do grupo multiponto e replicar a cápsula de dados para cada uma. Outra função das cápsulas de dados multipontos é verificar a cada nó ativo se existe um *cache* para o seu grupo multiponto. Caso exista, o nó ativo ao executar uma cápsula de dados multiponto deve copiar o seu conteúdo para o *cache*.

Cápsulas de pedido de retransmissão são enviadas sempre que perdas forem detectadas. Isto é feito através da verificação dos números de seqüência das cápsulas de dados multipontos. Qualquer salto no seqüenciamento é encarado como uma perda e cápsulas de pedido de retransmissão são enviadas requisitando o reenvio dos números de seqüência perdidos. Temporizadores são disparados nos receptores para garantir que se os números de seqüência requisitados não forem recebidos dentro de um intervalo de tempo baseado no RTT (*Round-time Trip*), novos pedidos de retransmissão serão enviados. Para este serviço, uma cápsula de pedido de retransmissão é enviada por número de seqüência perdido para minimizar a quantidade de processamento necessária nos nós ativos por pedido de retransmissão. Ao passar por um nó ativo, o procedimento desta cápsula deve verificar se existe o número de seqüência desejado no *cache* do grupo. Caso exista, o pedido de retransmissão é feito e a cápsula é descartada. Caso contrário, o procedimento deve marcar no *soft-state* do grupo o número de seqüência requisitado e prosseguir na direção do transmissor. As próximas cápsulas de pedido de retransmissão requisitando o mesmo número de seqüência serão descartadas. No entanto, a indicação de números de seqüência deve existir por um tempo limitado para que novos pedidos de retransmissão para novas perdas relacionadas ao mesmo número de seqüência não sejam impedidos de serem feitos.

A cápsula de fim de sessão existe para indicar o término de uma sessão para um grupo multiponto, o que significa o fim da comunicação para aquele grupo. Como o transmissor de um grupo multiponto não possui informações de controle sobre os usuários do seu grupo e nem do que foi recebido por eles, esta mensagem envia o último número de seqüência que foi transmitido. Desta forma, os receptores que não receberam as últimas cápsulas de dados multipontos podem requisitar a sua retransmissão. Para evitar que algum receptor não receba a cápsula de fim de sessão, cada receptor possui um temporizador de silêncio. Após um determinado tempo sem receber nenhuma cápsula do transmissor, o receptor envia um pedido de retransmissão para o último número de seqüência esperado. Ao receber este pedido, se o número de seqüência requisitado é maior do que o último enviado, o transmissor interpreta como sendo um pedido para a cápsula de fim de sessão e a retransmite.

3 Configuração do *Cache* de Cápsulas

O tamanho do *cache* para um grupo multiponto em um nó ativo utilizando o mecanismo de alocação adaptativa varia entre um valor mínimo e um valor máximo. A configuração do tamanho mínimo e máximo do *cache* é realizada durante a execução do procedimento de inscrição nos nós ativos intermediários, que recebe como parâmetro o RTT da cápsula. Desta forma, o cálculo destes tamanhos depende de fatores como: a distância do nó ativo ao receptor que enviou a cápsula de inscrição, a banda passante disponível (BW) e a taxa de perda de cápsulas. O tamanho mínimo é dado pela equação 1, enquanto que o tamanho máximo é dado pela equação 2.

$$Min = RTT \times BW + \frac{E[Data_{size}]}{Subscribe_{size}} \times RTT \times BW + TP \times BW - E[Data_{size}] \qquad (1)$$

O tamanho mímino corresponde a quantidade de bits que podem ser armazenados no intervalo de tempo entre os receptores detectarem a perda de uma cápsula de dados e a chegada do pedido de retransmissão no nó ativo. Este intervalo de tempo corresponde ao envio da cápsula de dados seguinte pelo nó ativo, seguido da sua recepção pelo membro do grupo, somado ao tempo de envio de um pedido de retransmissão e a subseqüente recepção deste pedido pelo nó ativo em questão. Além disto, deve-se considerar o tempo entre chegadas de cápsulas de dados multipontos e o tempo de processamento (TP). Desta forma, a equação para o tamanho mínimo contém quatro parcelas que consideram: o intervalo de tempo entre o envio de um pedido de retransmissão e a recepção deste pedido no nó ativo, considerando o tempo de envio de uma cápsula de dados e a recepção no membro do grupo, o tempo de processamento e o intervalo entre recepção de cápsulas de dados no nó ativo.

$$Max = K \times (Min + Retransmission_{interval} \times BW) \qquad (2)$$

$$P_{perda}^K \leq Const \tag{3}$$

O cálculo do tamanho máximo permitido pelo *cache* é dado pelo número de vezes consecutivas que um mesmo número de seqüência pode ser retransmitido e perdido por um nó ativo (K). Assim, o valor máximo depende do tamanho mínimo do *cache*, do intervalo entre pedidos de retransmissão para um mesmo número de seqüência e do valor de K. O valor de K depende da probabilidade de perda de uma cápsula de dados (P_{perda}) em uma determinada interface de saída de um nó. A equação 3 calcula o valor de K baseada na probabilidade de perda desejada e determinada pela aplicação ($Const$).

O valor do tamanho máximo e mínimo do *cache* de um grupo é calculado todas as vezes que uma cápsula de inscrição é recebida em um nó ativo. Os novos valores do máximo e do mínimo são comparados com os existentes e se forem maiores substituem os anteriores. Desta forma, usuários mais distantes do nó ativo não são prejudicados se aderirem mais tarde ao grupo multiponto.

$$\begin{aligned} RetBW \ &= \ (\#rets \times E[Data_{size}])/Intervalo \\ Se \quad & (RetBW > (F \times AvailBW)) \ e \ (Cache_{size} < Max) \\ & \quad Cache_{size} = Cache_{size} + (RetBW - (F \times AvailBW)) \times Intervalo \end{aligned} \tag{4}$$

Durante a transmissão de dados para o grupo multiponto, o tamanho do *cache* de um nó ativo varia de acordo com as condições da rede entre os tamanhos mínimo e máximo correntes. Esta variação depende do número de pedidos de retransmissão que não são atendidos no nó ativo e requisitam um número de seqüência nos níveis mais altos da árvore multiponto ($\#rets$). O procedimento das cápsulas de pedido de retransmissão é responsável por contabilizar o número de pedidos não atendidos e aumentar o tamanho do *cache* segundo a política determinada no algoritmo 4. A cada período de tempo igual a $Intervalo$, verifica-se a banda passante imposta pelas retransmissões geradas pelos pedidos não atendidos dentro deste período ($RetBW$). Se $RetBW$ for maior do que uma determinada fração (F, onde $0 < F \leq 1$) da diferença entre a banda passante disponível e utilizada ($AvailBW$), então aumenta-se o tamanho do *cache*. Observa-se que existe um compromisso entre a eficiência do mecanismo e a economia de recursos de memória em um nó ativo na escolha do valor de F, pois quanto menor for, mais freqüentemente o tamanho do *cache* é aumentado e portanto maior é o consumo médio de memória.

O tamanho do *cache* diminui suavemente conforme o número de pedidos de retransmissão não atendidos diminui. A cada $Intervalo$, se não houverem pedidos de retransmissão não atendidos, então o tamanho do *cache* é decrementado de um valor igual a $E[Data_{size}]$ (média do tamanho de uma cápsula de dados). Esta média é calculada pelo procedimento das cápsulas de dados multiponto a cada passagem de uma cápsula. O objetivo de diminuir suavemente o tamanho do *cache* é atingir um tamanho que mantenha a taxa de retransmissões não atendidas abaixo do nível que causaria congestionamentos nos enlaces dos níveis da árvore multiponto acima de um determinado nó ativo, mas sem utilizar recursos de memória em excesso para o *cache*.

4 Simulação

Para a obtenção de medidas foi utilizado o simulador de redes de computadores *ns* (*Network Simulator*), desenvolvido por pesquisadores do projeto VINT (*Virtual Internetwork Testbed*) [11]. Todo o suporte a redes ativas foi desenvolvido como extensão ao *ns*. Assim, nós ativos foram implementados para permitir a execução de procedimentos específicos das aplicações e um conjunto de primitivas foi disponibilizado para estes procedimentos, incluindo primitivas para a manipulação do *soft-state*. Para avaliar o serviço proposto, dois tipos de topologias foram utilizadas nas simulações: topologia em árvore cheia e topologia *transit-stub*.

O objetivo de realizar simulações utilizando topologias do tipo *transit-stub* é observar o comportamento do serviço proposto diante de estruturas semelhantes às que encontramos atualmente na Internet. Cada domínio na Internet pode ser classificado como sendo do tipo *stub* ou *transit* [12]. Em um domínio *stub* trafegam somente pacotes originados ou destinados aos nós do próprio domínio. Domínios do tipo *transit* não possuem esta restrição. Para obter este tipo de topologia, utilizou-se o gerador de topologias GT-ITM (*Georgia Tech Internetwork Topology Models*), que permite a geração de topologias aleatórias, mantendo o formato *transit-stub*.

4.1 Topologia em Árvore

Simulações em uma topologia em forma de árvore cheia foram realizadas com o objetivo de verificar a quantidade de retransmissões atendidas pelos nós ativos intermediários e a economia de recursos obtida quando utiliza-se o mecanismo de *cache* adaptativo ($Const = 0.05$, $F = 0.8$). Nestas simulações, considerou-se a raíz da árvore como sendo o transmissor do grupo multiponto, com uma fonte de tráfego CBR de $533 kbits/s$ (cápsulas de dados de 1000 bytes com intervalos de $0.015s$), e as folhas são os nós aos quais estão ligados os receptores, sendo que o grau da árvore (número de filhos máximo de um nó da árvore) escolhido foi três (3), pois fornece uma dispersão suficiente dos usuários do grupo multiponto para as medidas desejadas. A topologia utilizada está ilustrada na figura 2.

Para cada simulação, a posição dos membros do grupo foi escolhida aleatoriamente, tendo sido mantido constante o número de membros do grupo em 40. Para estes usuários, assume-se uma banda passante disponível de 1 Mbit/s, sendo este o valor atribuído a cada enlace. Perdas aleatórias foram consideradas para simular erros nos enlaces e congestionamentos, utilizando-se taxas de erro de 20%. No entanto, estas taxas de erro não foram associadas simultaneamente a todos os enlaces. Simulações foram realizadas considerando-se estas taxas de erro em apenas um nível por vez, ou seja, como está ilustrado na figura 2, apenas os enlaces de um dos quatro níveis marcados pelas linhas pontilhadas possuem taxas de erro de 20%, associando aos restantes taxas de erro de 0.25%. Os resultados para estas simulações estão apresentados nas figuras 3, 4 e 5.

A figura 3 apresenta quatro gráficos que mostram a proporção de pedidos de retransmissão atendidos nos nós ativos de cada nível da árvore, sendo que cada gráfico considera taxas de erro de 20% em um nível diferente. Além das medidas utilizando *cache* adaptativo, foram realizadas medidas considerando alocações pelo mínimo e pelo máximo. Nestas medidas, o tamanho do *cache* não varia entre o mínimo e o máximo calculados, isto é, o tamanho é sempre alocado pelo mínimo ou pelo máximo. Observa-se nestes gráficos que nos níveis mais baixos (0 e 1), como existem mais enlaces gerando perdas aleatoriamente,

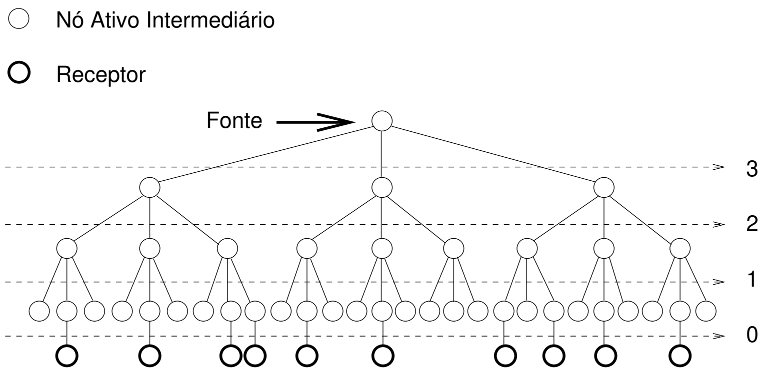
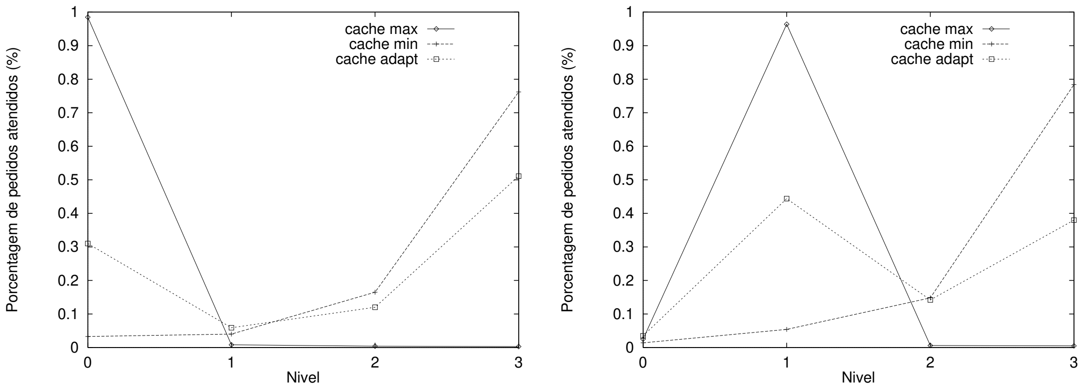
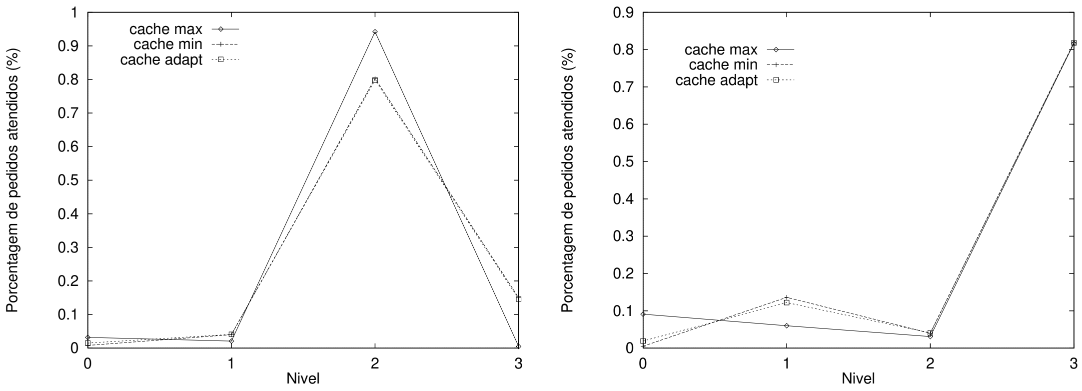


Figura 2: Árvore cheia com grau 3.



(a) Nível 0.

(b) Nível 1.



(c) Nível 2.

(d) Nível 3.

Figura 3: Retransmissões na topologia em árvore.

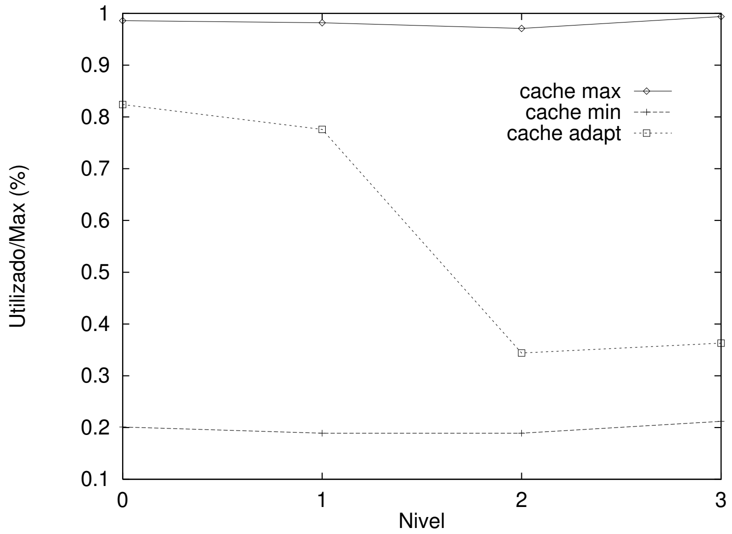


Figura 4: Utilização do *cache* com relação ao tamanho máximo negociado.

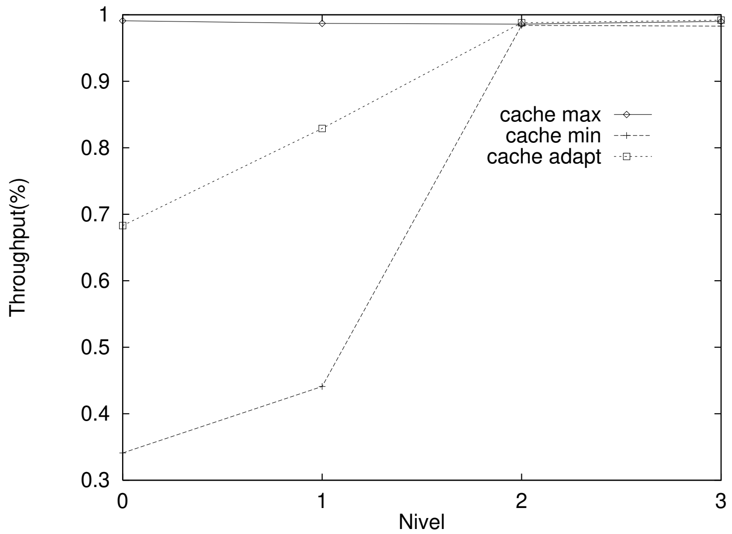


Figura 5: Vazão obtida/Vazão nominal da fonte.

mais pedidos de retransmissão para números de seqüência diferentes serão enviados, ocasionando um aumento no uso de recursos de memória nos nós ativos dos níveis mais altos que recebem uma grande quantidade de pedidos de retransmissão, para o caso de *cache* adaptativo. No entanto, para os nós ativos dos níveis mais baixos, como o número de pedidos de retransmissão não é elevado, a quantidade de recursos utilizada não é grande e, por conseqüência, o número de pedidos atendidos é menor, comparando com a alocação pelo máximo. Para os dados dos níveis mais altos (2 e 3), observa-se que as três curvas se aproximam, mostrando que a quantidade de recursos de memória necessária é menor devido ao menor número de pedidos de retransmissão para números de seqüência diferentes. Assim, o mecanismo de alocação adaptativa economiza recursos nesta situação.

O gráfico apresentado na figura 4 apresenta a proporção média de utilização do cache com relação ao máximo para os três casos e para taxas de erro em cada nível. Para o *cache* adaptativo, observa-se que para os níveis mais baixos a utilização de memória é maior, mas, ainda assim, não atinge o valor máximo. Por outro lado, nos níveis mais altos, a utilização de memória pelo *cache* se aproxima do mínimo. Desta forma, a economia média de recursos utilizando *cache* adaptativo para este caso varia entre 28-66 %.

O gráfico de vazão da figura 5 mostra que para os níveis mais baixos, como a utilização de banda passante é maior, a vazão para o *cache* adaptativo diminui, mas mantém um bom desempenho e economiza na média mais de 28% de recursos de memória se comparado à alocação pelo máximo.

4.2 Topologia ***Transit-Stub***

A topologia da figura 6 foi utilizada na obtenção dos resultados apresentados nesta seção. Nesta topologia, pode-se observar o formato de um núcleo (*transit*) interligando várias sub-redes (*stubs*). Para as simulações realizadas nesta topologia, a banda passante dos enlaces gerados pelo GT-ITM e a taxa média de perdas foram consideradas como sendo $1.5 Mbits/s$ e 10%, respectivamente, enquanto que os atrasos de propagação (proporcional ao comprimento do enlace) foram gerados aleatoriamente pelo GT-ITM, estando compreendidos no intervalo $[1.1ms, 4.6ms]$.

Os receptores do grupo multiponto são nós ligados às sub-redes. Um número fixo de cinco (5) usuários por sub-rede foi determinado, escolhendo-se aleatoriamente o nó da sub-rede ao qual cada usuário está ligado, totalizando um total de 40 usuários. Os enlaces que ligam os receptores aos nós da sub-rede possuem a mesma banda passante de $1.5 Mbits/s$, mas com atraso de propagação de $0.1ms$ e taxas médias de erro de 20%. Uma taxa média de erro mais elevada foi considerada para os enlaces que ligam os receptores para averiguar o comportamento do mecanismo proposto diante de enlaces sem fio, que possuem como característica elevadas taxas de perdas.

Para avaliar o funcionamento do *cache* adaptativo ($Const = 0.05$, $F = 0.8$), medidas foram realizadas utilizando uma fonte de tráfego modelada por um processo de Poisson. Esta fonte foi posicionada em um nó do núcleo, escolhido aleatoriamente, tendo sido variada a taxa média de geração de dados. Medidas para comparação foram igualmente realizadas considerando-se alocações pelo máximo e pelo mínimo.

O gráfico da figura 7 apresenta a razão entre a vazão média dos receptores e a vazão média da fonte de tráfego obtida para os três mecanismos de alocação de *cache*. Observa-se por este gráfico que até taxas médias de geração de dados de $200kbits/s$, os três mecanismos atingem a vazão máxima. Neste caso, como pode ser observado no gráfico da

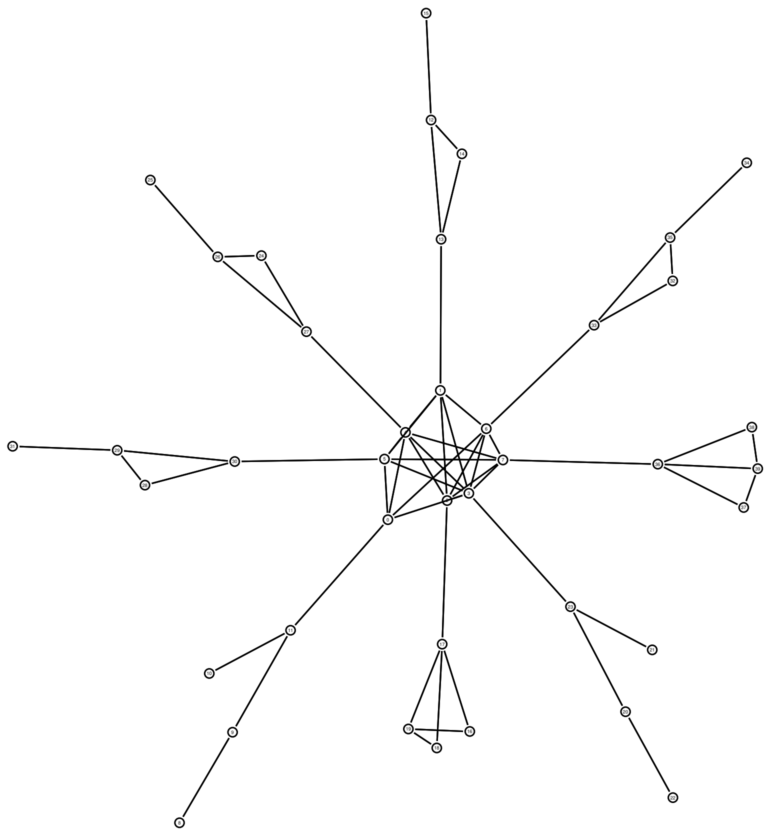


Figura 6: Topologia *transit-stub*.

figura 8 que mostra a razão entre a média do tamanho do *cache* e o tamanho máximo calculado, a alocação pelo máximo desperdiça recursos, pois o *cache* adaptativo obtém o mesmo resultado utilizando uma quantidade próxima do mínimo. Entre $200kbits/s$ e $900kbits/s$, a vazão cai, sendo que a alocação pelo máximo mantém a maior vazão, pois aloca previamente a quantidade máxima. Para as taxas médias de geração neste intervalo, a quantidade de recursos utilizada pelo *cache* adaptativo cresce até saturar em $533kbits/s$, que ocorre devido ao tamanho do *cache* que atinge o máximo. A partir deste mesmo valor, o uso de *cache* para a alocação pelo máximo decresce devido ao surgimento de congestionamentos no sentido da fonte causados pelo grande número de pedidos de retransmissão e o conseqüente aumento nos valores do *cache* mínimo e máximo calculados, que acarretam num valor médio menor do que o máximo valor observado. Para taxas médias de geração acima de $900kbits/s$, a vazão obtida utilizando *cache* adaptativo e alocação pelo máximo são aproximadamente iguais, utilizando desta forma a mesma quantidade de recursos de memória para o *cache*.

A figura 9 mostra a distribuição do atraso dos pacotes retransmitidos durante as simulações realizadas. A alocação pelo máximo, por alocar uma quantidade maior de recursos mesmo quando não há necessidade, consegue realizar retransmissões mais rápidas para uma maior parte da amostra (figura 9.a). Ao contrário, a alocação pelo mínimo, por alocar sempre o mínimo de recursos necessários, apresenta maiores atrasos nas retransmissões (figura 9.b). A utilização do mecanismo de *cache* adaptativo procura manter o *cache* com um tamanho capaz de realizar retransmissões o mais rápido possível, mas mantendo o compromisso de não desperdiçar recursos. Assim, a distribuição do atraso

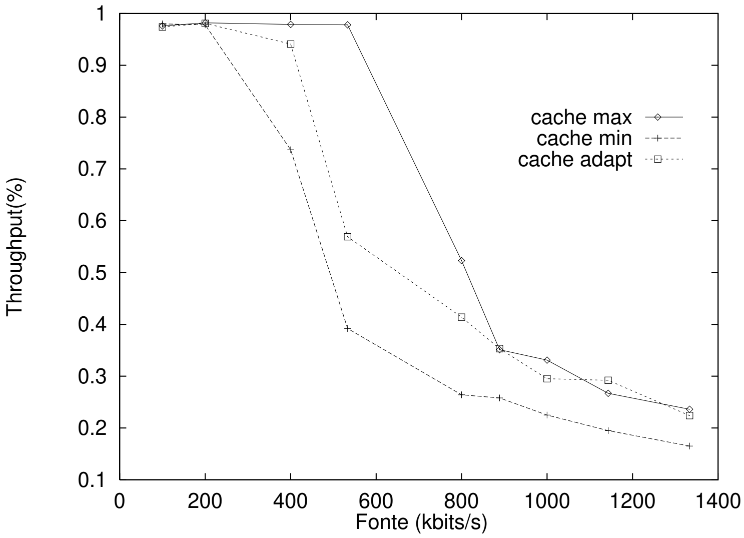


Figura 7: Vazão x Taxa média de geração da fonte.

para as retransmissões apresentada na figura 9.c se encontra entre os casos de alocação pelo máximo e alocação pelo mínimo.

5 Conclusões

Este artigo apresentou um serviço de comunicação multiponto confiável baseado em redes ativas. *Caches* de cápsulas de dados foram utilizados nos nós ativos intermediários com o objetivo de realizar retransmissões mais rápidas e dentro do escopo onde ocorreram as perdas, diminuindo assim o atraso das retransmissões e o consumo de banda passante. Para a alocação do *cache* foi proposto um mecanismo adaptativo capaz de adequar este tamanho às condições correntes da rede. Os principais objetivos foram a economia de recursos de memória, quando a taxa de perdas é baixa, e o máximo possível de retransmissões localizadas, quando esta taxa aumenta.

Os resultados obtidos por simulação demonstraram o bom desempenho do serviço utilizando o mecanismo de *cache* adaptativo. As simulações realizadas na topologia em árvore mostraram o efeito do aumento da taxa de perdas em diferentes posições na árvore. Perdas ocorridas nas posições mais altas da árvore multiponto geram pedidos de retransmissão simultâneos para os mesmos números de seqüência, mantendo o *cache* próximo ao mínimo e, portanto, economizando recursos de memória. Se as perdas ocorrerem nas posições mais baixas, pedidos de retransmissão para números de seqüências diferentes são gerados, aumentando a utilização de recursos de memória e, conseqüentemente, o número de retransmissões localizadas.

Na topologia *transit-stub*, avaliou-se o desempenho do serviço em termos da vazão obtida para os receptores mudando as características da fonte de tráfego e considerando altas taxas de erro. Para fontes de tráfego com taxas médias de até $200kbits/s$, a vazão máxima foi obtida, mantendo a utilização de recursos de memória próxima do mínimo. Acima deste taxa de geração para a fonte, a vazão dos receptores diminui e o consumo de recursos aumentou, até saturar em $533kbits/s$, quando atingiu o consumo máximo de memória.

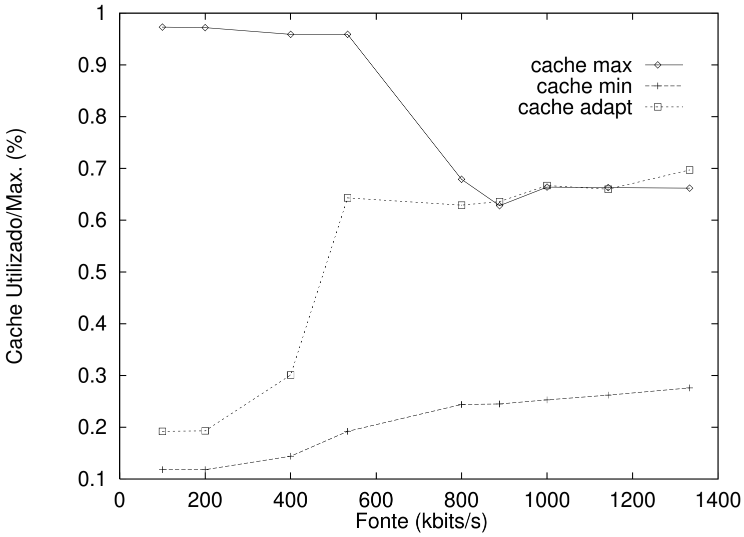
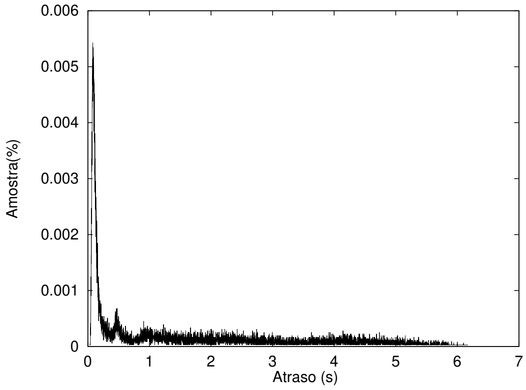
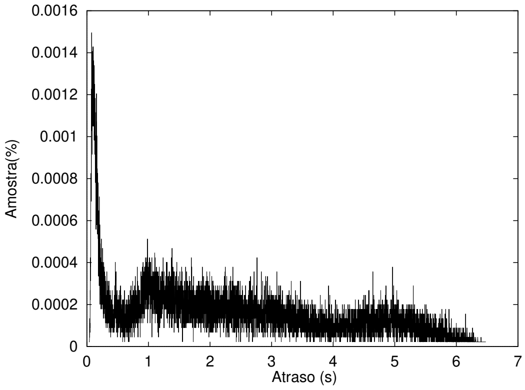


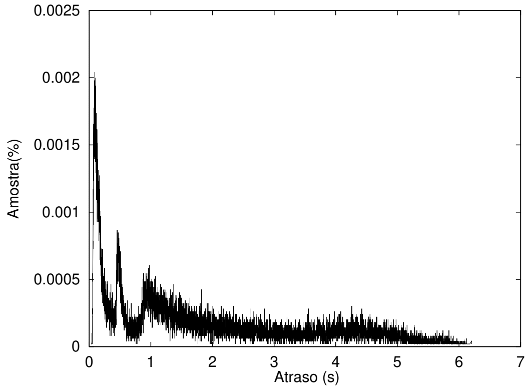
Figura 8: Utilização do *cache* com relação ao máximo negociado.



(a) Alocação pelo máximo.



(b) Alocação pelo mínimo.



(c) Cache Adaptativo.

Figura 9: Distribuição dos atrasos das retransmissões.

Portanto, o uso do mecanismo de alocação de *cache* adaptativo mostrou-se eficiente para aplicações que precisam garantir a presença de *cache* de dados em nós intermediários, pois evitou o desperdício de recursos de memória e garantiu um bom desempenho geral do serviço, mesmo tendo sido consideradas altas taxas de erro.

Referências

[1] S. Floyd, V. Jacobson, C. Liu, S. McCanne e L. Zhang, "A Reliable Multicast Framework for Light-weight Sessions and Application Level Framing", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 784–805, dezembro 1997.

[2] D. Towsley, J. Kurose e S. Pingali, "A Comparison of Sender-Initiated and Receiver-Initiated Reliable Multicast Protocols", *Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 398–406, abril 1997.

[3] C. Diot, W. Dabbous e J. Crowcroft, "Multipoint Communication: A Survey of Protocols, Functions and Mechanisms", *Journal on Selected Areas in Communications*, vol. 15, no. 3, pp. 277–290, abril 1997.

[4] L. H. Lehman, S. J. Garland e D. L. Tennenhouse, "Active Reliable Multicast", in *IEEE INFOCOM'98*, San Francisco CA, EUA, abril 1998.

[5] D. Wetherall, U. Legedza e J. Guttag, "Introducing New Internet Services: Why and How", *IEEE Network*, vol. 12, no. 3, pp. 12–19, junho 1998.

[6] D. J. Wetherall, J. V. Guttag e D. L. Tennenhouse, "ANTS: A Toolkit for Building and Dynamically Deploying Network Protocols", in *IEEE OPENARCH'98*, San Francisco CA, EUA, abril 1998.

[7] K. Calvert, E. W. Zegura, S. Bhattacharjee e J. Sterbenz, "Directions in Active Networks", *IEEE Communications Magazine*, vol. 36, no. 10, pp. 72–78, outubro 1998.

[8] M. Calderon, M. Sedano, A. Azcorra e C. Alonso, "Active Network Support for Multicast Applications", *IEEE Network*, vol. 12, no. 3, pp. 46–52, junho 1998.

[9] U. Legedza, D. J. Wetherall e J. Guttag, "Improving the Performance of Distributed Applications Using Active Networks", in *IEEE INFOCOMM'98*, San Francisco CA, EUA, abril 1998.

[10] F. P. Junqueira e O. C. M. B. Duarte, "Comunicação Multiponto para Estações Móveis em Redes Ativas", a ser apresentado no *XVII Simpósio Brasileiro de Redes de Computadores*, Salvador, BA, maio 1999.

[11] K. Fall e K. Varadhan, *NS Notes and Documentation.* The VINT Project, novembro 1997.

[12] E. W. Zegura, K. L. Calvert e M. J. Donahoo, "A Quantitative Comparison of Graph-Based Models for Internet Topologies", *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 770–783, dezembro 1997.