

Memória Cooperativa Distribuída para Sistemas de VoD peer-to-peer

Edison Ishikawa^(1,2)
edsoni@cos.ufrj.br

Cláudio Amorim⁽¹⁾
amorim@cos.ufrj.br

⁽¹⁾COPPE – Programa de Engenharia de Sistemas e Computação
Universidade Federal do Rio de Janeiro
Centro de Tecnologia, bloco H-318, Ilha do Fundão, 21945-970
Rio de Janeiro - RJ

⁽²⁾ Instituto Militar de Engenharia – IME

Resumo

Sistemas de Vídeo sob Demanda (VoD) convencionais são limitados pela banda passante de E/S do servidor bem como pela banda passante da rede. Neste artigo, introduzimos uma nova técnica denominada Memória Cooperativa Distribuída (MCD) que ataca tais gargalos, possibilitando a implementação de sistemas de VoD interativos e escaláveis. A idéia chave por trás da MCD é o uso dos buffers dos clientes como uma única memória global distribuída de tal forma que seu uso diminui substancialmente os requisitos de banda passante de o sistema. Em contraste com os sistemas convencionais cliente/servidor de VoD, MCD implementa um sistema peer-to-peer de VoD, em que cada cliente é também um potencial servidor de VoD. Através de detalhadas simulações, nossos resultados preliminares mostram que sistemas de VoD baseados em MCD são potencialmente escaláveis. Estes resultados sugerem que a MCD pode ter um papel promissor no desenvolvimento de sistemas de VoD interativos e escaláveis.

Abstract

Conventional Video-on-Demand (VoD) systems are limited by the I/O server bandwidth as well as the network bandwidth. In this paper, we introduce a new technique called Distributed Cooperative Memory (MCD) that addresses such bandwidth bottlenecks so as to enable VoD systems to become scalable and interactive. The key insight behind MCD is to use the client buffers as a single global distributed memory in such a way that it can reduce dramatically the bandwidth requirements of VoD systems. In contrast with the conventional client/server VoD system, MCD implements a peer-to-peer VoD system, in which each client is also a potential VoD server. Through detailed simulations, our preliminary results show that MCD-based VoD systems are potentially scalable. These results suggest that MCD may have a promising role in the development of scalable and interactive VoD systems.

Palavras-chave: vídeo sob demanda, sistemas multimídia distribuídos, gerenciamento de memória global.

1 Introdução

Apesar de a aparente simplicidade do serviço de Vídeo sob Demanda [5] (VoD) convencional, onde o usuário é fortemente motivado pela interatividade do sistema, o custo de se implementar um sistema de VoD escalável e interativo não compensa frente a sistemas de distribuições de vídeo mais econômicas como o aluguel de fitas ou o sistema “pay-per-view”. Isto porque o sistema de VoD convencional não usa a banda passante de forma eficiente, dedicando um fluxo de vídeo para cada cliente, o que exige um servidor de VoD e uma infraestrutura de comunicações dispendiosos.

A questão básica é escalar o sistema para suportar um grande número de usuários sem precisar crescer a capacidade do servidor linearmente com o número de usuários. Normalmente as soluções propostas escalam o sistema, mas este deixa de oferecer todas as facilidades de operação de videocassete (VCR), por usar não apenas um fluxo por usuário, mas um fluxo multicast [9, 11].

Para contornar este problema, estamos propondo um modelo alternativo para o sistema de VoD, o modelo "peer-to-peer", bastante popularizado com o advento de programas como o Napster [10] e Gnutella [1] para citar alguns, ao invés do tradicional modelo cliente/servidor, onde cada cliente VoD é também um potencial servidor de VoD. Para tornar isto possível, desenvolvemos o conceito de Memória Cooperativa Distribuída (MCD). A MCD aproveita a idéia usada em Chaining [7] para encadear os buffers de clientes formando uma linha de retardo. Com este encadeamento aproveita-se melhor o fluxo que vai para o cliente e passa por seu buffer que, ao invés de ser descartado é repassado a algum cliente que o esteja necessitando. Como este procedimento pode ser repetido indefinidamente, consegue-se formar longas cadeias de buffer dos clientes por onde o fluxo de vídeo passa. Desta forma a enorme quantidade de informação bufferizada no sistema pode ser aproveitada para gerar novos fluxos de vídeo.

A inovação da MCD consiste na utilização do espaço de armazenamento dos buffers não apenas como uma linha de retardo, mas como uma área coletiva de memória permitindo que esta seja usada de forma cooperativa pelo sistema de VoD como um todo, oferecendo facilidades de operações de videocassete, sempre de maneira escalável. MCD é flexível, pois permite atender desde pedidos de filmes muito populares, situação em que seu desempenho é maximizado, até pedidos de filmes pouco vistos onde seu uso não afeta o desempenho do sistema de VoD. A economia que pode ser conseguida com a MCD é significativa se levarmos em conta que a maioria dos usuários assiste aos filmes ditos populares e numa mesma faixa de horário, o horário nobre da televisão.

A seguir a seção 2 descreve as técnicas que exploram os recursos do cliente em benefício do sistema de VoD com o objetivo de tornar o texto autocontido. A técnica proposta, a MCD, é descrita na seção 3. As simulações são discutidas na seção 4, os trabalhos relacionados na seção 5 e por fim as considerações finais e trabalhos futuros são apresentados na seção 6.

2 Explorando os recursos do cliente VoD

Explorar os recursos do cliente implica em se poder usar a capacidade de armazenamento do cliente bem como a sua capacidade de não só receber informações, mas também retransmití-las, com o objetivo de aumentar a escalabilidade do sistema.

Uma forma de usar estes recursos em um sistema de VoD é através do compartilhamento do fluxo de vídeo transmitido. O multicast é uma forma de compartilhar este fluxo, mas ele é um recurso da rede de transmissão. Um novo cliente pode explorar um fluxo multicast já existente (figura 1 – fase 1) se juntando a um grupo de multicast que esteja recebendo o vídeo escolhido (figura 1 – fase 2). Desta forma ele compartilha um fluxo já existente, o fluxo normal, economizando recursos do servidor VoD. No entanto, todo novo cliente quer ver um vídeo desde o começo, ou seja, o fluxo que ele está recebendo só será útil mais à frente, de forma que ele armazena localmente este conteúdo e solicita a parte que estava faltando ao servidor, o *patch* ou remendo (figura 1 – fase 3), donde o nome da técnica *Patching* [3]. Note que o cliente usa sua capacidade de armazenamento local para

posteriormente reaproveitar o conteúdo de um fluxo existente e, ao mesmo tempo, explora a banda passante excedente de sua placa de rede para receber dois fluxos de vídeo simultâneos, o fluxo normal e o *patch*. Este esquema é melhor porque o *patch* usa os recursos do servidor por menos tempo do que um novo fluxo convencional. MCD também usa o *patch*, mas o *patch* não é necessariamente fornecido pelo servidor, ele pode ser fornecido, e o é, preferencialmente, pelo cliente.

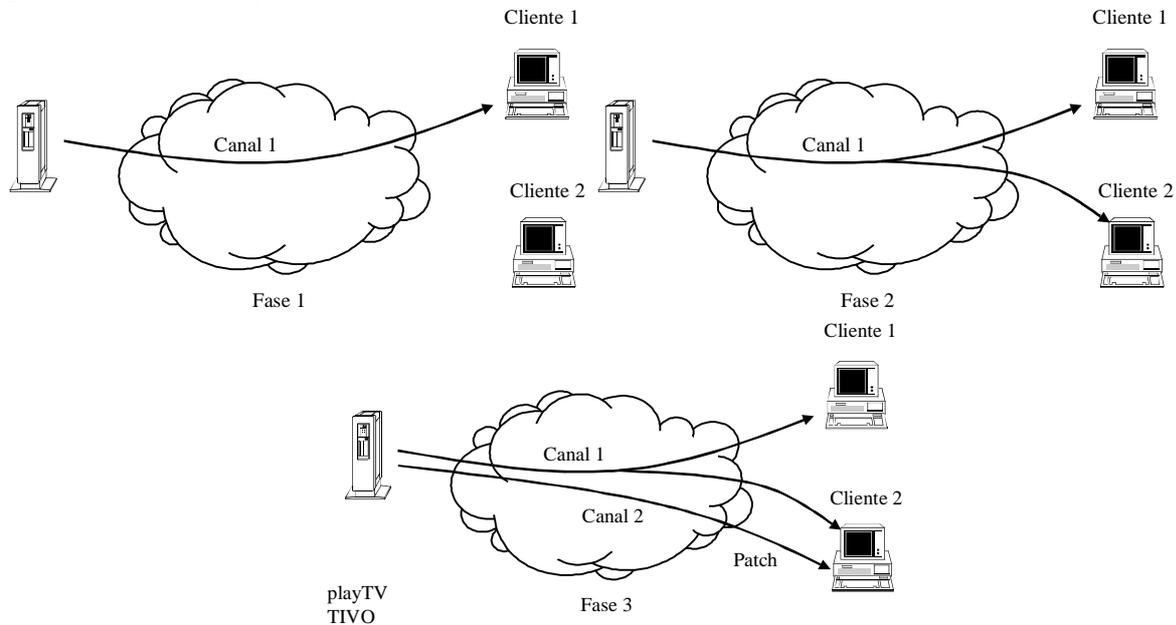


Figura 1: Patching – mecanismo em três fases

Uma outra forma de usar os recursos do cliente é através da retransmissão do conteúdo que ele recebeu e ainda está armazenado localmente. Normalmente o conteúdo recebido é descartado após a sua utilização, a idéia é reciclá-lo. Isto é feito da seguinte forma: o primeiro cliente a chegar no sistema é servido por um novo fluxo do servidor. Se o segundo cliente chegar ao sistema antes que o primeiro cliente descarte a parte inicial do fluxo do seu buffer, então o primeiro cliente pode fornecer um fluxo reciclado para o segundo cliente e assim por diante. Caso contrário, o servidor inicia um novo fluxo para o segundo cliente. Este esquema cria uma longa cadeia (*chain*) de buffers, por isso seu nome Chaining [1]. MCD também usa o conceito de Chaining, mas com o cuidado de minimizar o tráfego de um cliente para o outro que pode ficar cruzando a rede e eliminado o problema de algum cliente interromper a cadeia por qualquer motivo. A figura 2 apresenta uma cadeia de *buffers* de Chaining.

3 Memória cooperativa distribuída

A MCD é um nova maneira de gerenciar a memória global formada pelos buffers dos clientes do Sistema de VoD. Para acessar partes do conteúdo do vídeo na memória coletiva usa-se o GoF (*Group of Frames*) como unidade de acesso. O GoF, definido no padrão MPEG [6], é composto por uma sequência de quadros, por exemplo IBBPBBPBBPBBP, onde o quadro I é o quadro Intracodificado. O quadro I é gerado pela compressão de uma cena completa, neste sentido, este quadro é autocontido, pois não precisa da informação de outros

quadros para ser decodificado. O Quadro P é o quadro Preditivo; este quadro contém a predição do movimento de um quadro I ou P anterior. Logo, para decodificá-lo é preciso a informação dos frames anteriores. O quadro B é o Bidirecional, ele contém a diferença entre o último e o próximo quadro I ou P, portanto para decodificá-lo precisa-se da informação destes quadros. Da mesma maneira, o GoF também é autocontido e tem um *timestamp*, de modo que se pode acessá-lo como uma palavra em uma RAM.

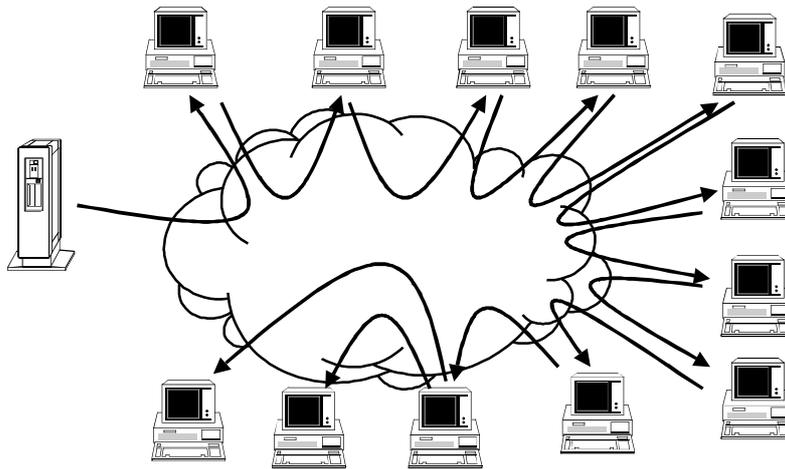


Figura 2: Chaining

Um cliente de VoD precisa uma quantidade mínima de buffer para lidar com a variação estatística do retardo (jitter) da rede. A MCD estende o controle sobre este buffer para implementar um protocolo que permita o compartilhamento do seu conteúdo com outros clientes.

O buffer é dividido da seguinte forma:

- Início – aponta para o início do GoF com o menor *timestamp*. À medida que os GoFs são descartados ele é atualizado.
- Leitura – aponta para o início do próximo GoF a ser exibido pelo cliente.
- Escrita - aponta para a posição onde o próximo GoF a ser recebido deve ser armazenado.
- Mínimo – indica quando o buffer está para atingir o ponto de *underflow*, gerando um sinal para o gerente MCD tomar as medidas necessárias para iniciar um novo fluxo para recompletar o buffer.
- Máximo – indica a possibilidade de *overflow* no buffer. Quando o buffer está para atingir este nível, o cliente gera um sinal para o gerente MCD que irá tomar as ações necessárias para prevenir um *overflow* e conseqüente perda de GoFs.

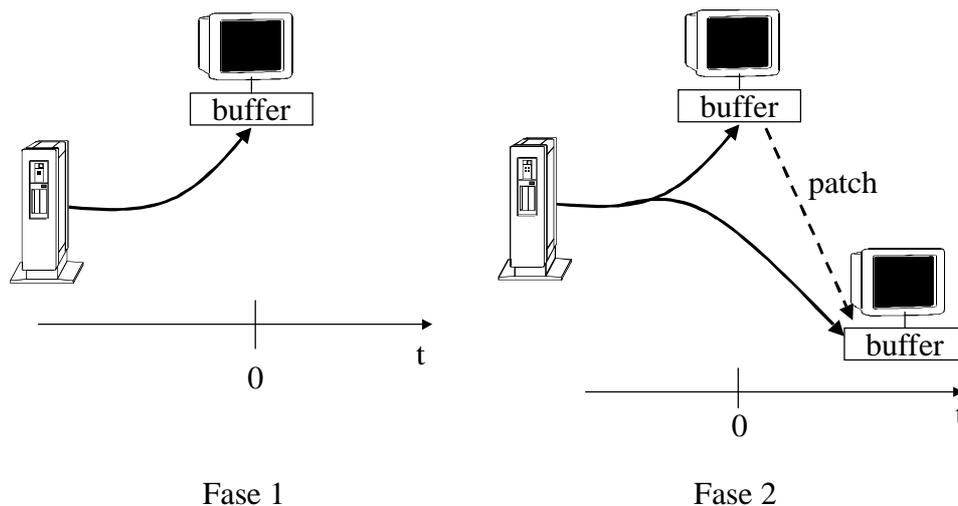


Figura 3: O patch na MCD

Para se entender o funcionamento da MCD, considerar-se-á, apenas por questões de simplificação e melhor entendimento, um servidor com apenas um filme armazenado.

O primeiro cliente que chega no sistema pede o filme escolhido ao gerenciador MCD. Como não há outro fluxo de vídeo, o gerente MCD libera o servidor VoD para que este forneça um novo fluxo de vídeo (figura 3-fase 1). Quando o cliente atinge o nível mínimo, ele começa a exibir o filme. Até este ponto MCD é como um sistema de VoD convencional.

O próximo cliente a chegar, chamado de requisitante, pede um filme e o gerente MCD procura na tabela MCD um cliente que tenha a parte inicial do filme no seu buffer, doravante chamado de provedor. Seja t_b a quantidade média de vídeo armazenada no buffer do cliente em segundos. Ele pode ser calculado dividindo-se o tamanho do buffer pela vazão média do fluxo de vídeo. Como é uma média, o tamanho estimado é reduzido para dar uma folga maior ao sistema, não considerando o espaço deixado para evitar o overflow. Se o cliente procurado na tabela recebeu o primeiro GoF do filme t_b segundos antes da tempo de requisição, então ele é um candidato em potencial para ser o provedor. Note que a base de tempo usada pelo cliente é quando ele recebe o primeiro GoF do vídeo, este momento é denominado tempo inicial relativo (figura 3-fase 2). Se existir mais de um cliente nesta condição, o gerente MCD usa as seguintes regras para escolher o novo provedor:

- Se existe um cliente cujo tempo inicial relativo é $t_b/2$ segundos menor que o tempo corrente e ele já é um provedor, então ele também será o provedor, colocando o novo cliente no seu grupo de multicast já existente. A parte inicial perdida é fornecida por um outro cliente do grupo de multicast, o colaborador, que manda um patch com a parte perdida do filme para o requisitante. (Este tamanho, $t_b/2$, é um compromisso entre explorar o fluxo multicast ao máximo com a necessidade de se fazer *patches* de tamanho longo ou fazer *patches* curtos mas diminuindo o número de usuários no grupo de multicast. Note também que o sistema pode mandar os *patches* em rajadas e *patches* longos usam mais banda passante da rede, o que aumenta o tráfego na rede, mas enche o buffer do cliente mais rapidamente, reduzindo a latência para a exibição do filme.)
- Senão, o último cliente que recebeu o fluxo de vídeo até t_b segundos é o novo provedor, estendendo a cadeia em mais um buffer.

- Caso não encontre nenhum cliente nas condições acima, libera o servidor para fornecer um novo fluxo de vídeo.

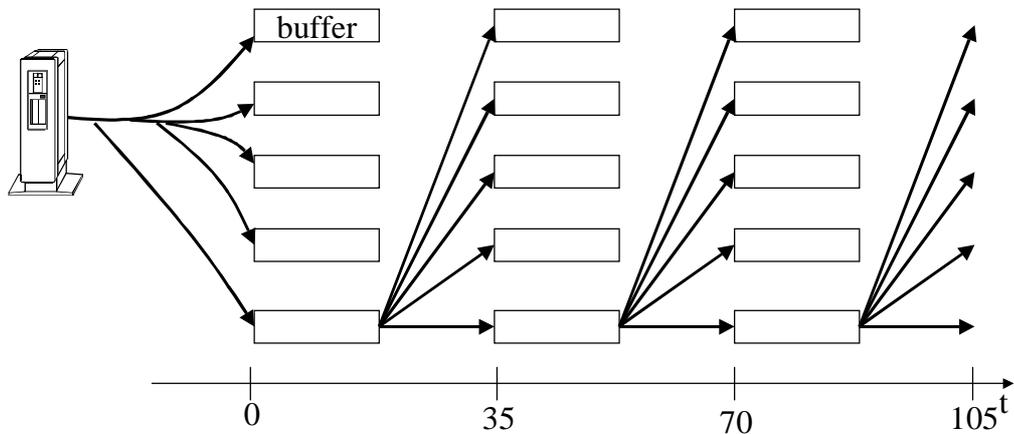


Figura 4: Árvore de buffers formada pela MCD

Note que em Patching o remendo é sempre enviado pelo servidor e na MCD ele é otimizado para ser mandado preferencialmente pelo cliente colaborador. Outro detalhe importante da MCD é que cada cliente provedor fornece apenas um fluxo multicast, ao contrário de Chaining, o que leva a um menor congestionamento da rede. O restante da banda passante de subida é usada para o protocolo de controle da MCD, para os *patches* e operações de avanço e retrocesso rápido.

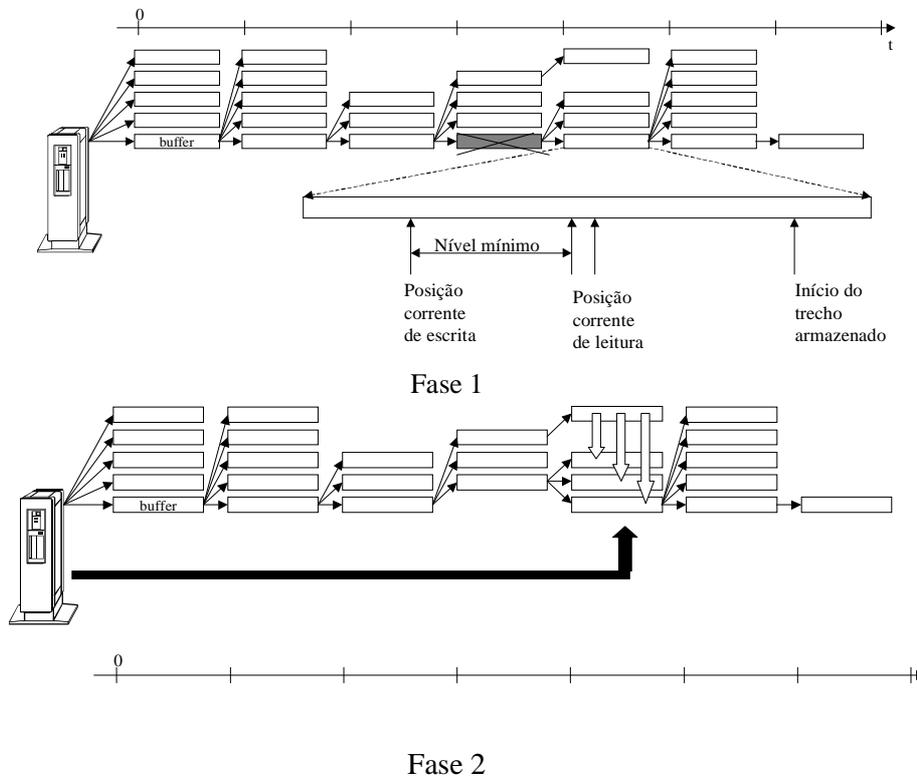


Figura 5: Tratamento da falha silenciosa na MCD

A figura 4 apresenta uma árvore de buffers gerada pela MCD, no exemplo o buffer tem um tamanho suficiente para armazenar em média 80 segundos de vídeo, com um nível máximo em 70 segundos. No caso o valor de $tb/2$ é $70/2$ o que leva o sistema a ter slots de 35 segundos.

Se qualquer provedor do sistema falhar silenciosamente, o requisitador irá consumir os GoFs existentes em seu buffer até atingir o nível mínimo. Neste momento, o requisitante receberá um alerta do buffer e irá acionar o gerente MCD para procurar outro provedor (figura 5 – fase 1) para mandar o fluxo de vídeo. Enquanto isso, o requisitante ainda tem GoFs a serem exibidos, dando tempo para providenciar um novo fluxo. O novo provedor pode ser um outro cliente do grupo de multicast, ou na falta deste o próprio servidor. Se o novo provedor também já era um provedor, seu cliente pode mandar patches, caso necessário (figura 5 – fase 2). Este tratamento, embora simples, resolve o problema da interrupção da cadeia existente em Chaining.

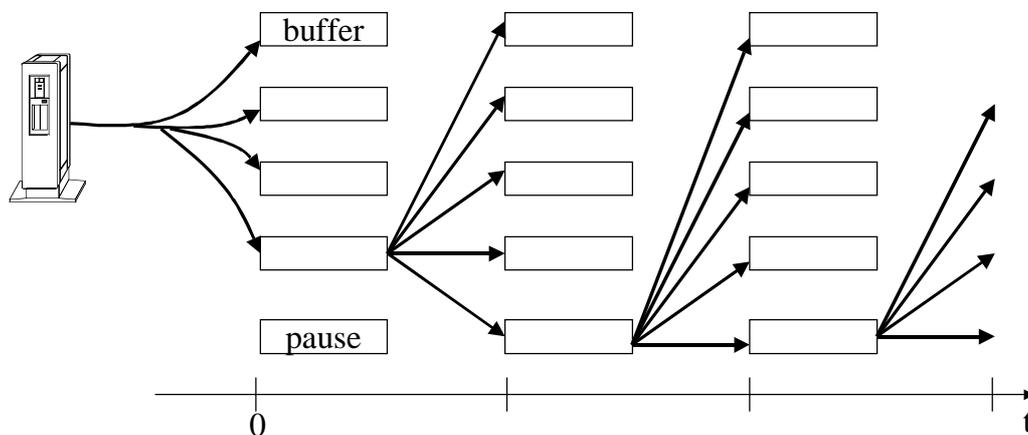


Figura 6: Pausa na MCD

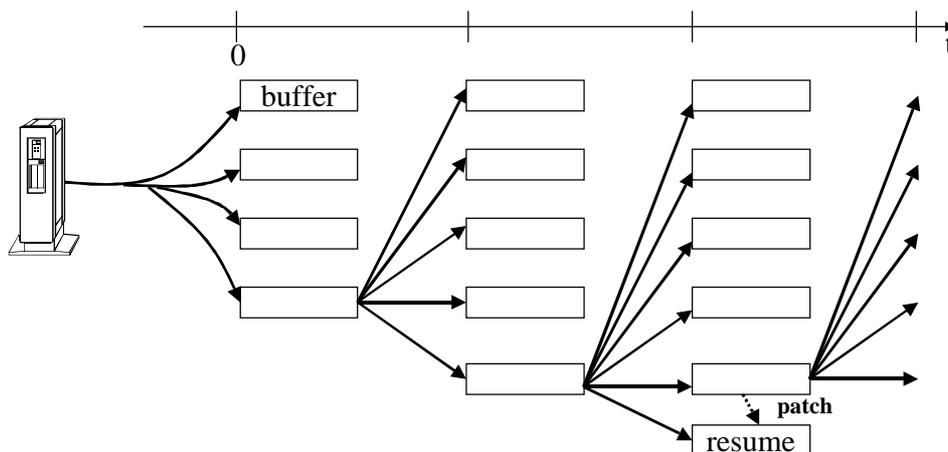


Figura 7: Reinício na MCD

A operação de pausa na MCD também é simples. Se o cliente que requisitou a pausa é uma folha da árvore, então a solução é trivial. Ele apenas notifica o gerente MCD que providenciará a retirada do mesmo do grupo de multicast do seu provedor, deixando de receber o fluxo de

vídeo. Se o cliente a fazer a pausa for um nó da árvore, um provedor, o gerente MCD irá percorrer esta árvore uma aresta em direção à raiz achando o provedor do provedor. Se este provedor do provedor tiver outros clientes no seu grupo de multicast, ele escolhe um para ser substituto do provedor que pediu a pausa (A figura 6 é o estado da árvore após a pausa, o estado anterior é ilustrado pela figura 4). Note que os clientes de um mesmo grupo de multicast tem o mesmo conteúdo em seu buffer e por isso são os substitutos perfeitos do provedor que pediu a pausa. No caso de não existir outro cliente no grupo de multicast, o gerente MCD procura outro provedor em sua tabela como numa requisição inicial, descrita anteriormente.

Quando um cliente em pausa manda um comando de reiniciar para o gerente MCD, ele inicia a exibição com os GoFs existentes em seu buffer e, enquanto isso, o gerente MCD percorre os nós da árvore na mesma direção da fluxo do vídeo até achar um provedor que também tenha o mesmo tempo decorrido do início da exibição do filme. Note que o cliente que reiniciou a exibição tem um novo tempo relativo de início, dado pelo eixo do tempo na figura 7, o tempo corrente menos o tempo do timestamp do último GoF recebido. Se a árvore terminar antes que o gerente MCD ache um novo provedor, então o servidor fornece um novo fluxo para atender este pedido de reinício.

Usando a mesma lógica de se percorrer os nós da árvore formada pelos buffers dos clientes é possível implementar operações de avanço e retrocesso rápido (ff e rw respectivamente), existentes nos aparelhos de videocassete, usando-se a MCD. Estas operações podem ser implementadas tanto de forma discreta como contínua.

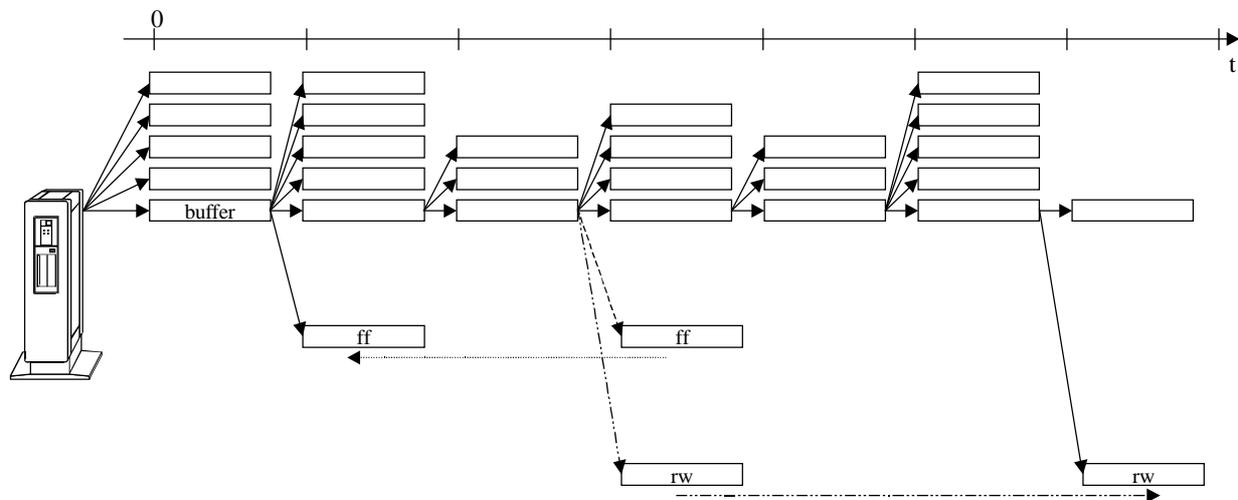


Figura 8: Avanço e retrocesso rápido discreto

Para implementar operações de ff/rw discretas usa-se a vazão média do fluxo de vídeo para se calcular quantos GoFs aproximadamente são necessários para avançar ou recuar até o ponto requisitado. Se este movimento for pequeno, os GoFs necessários estarão no buffer local. Do contrário, o cliente pede ao gerente MCD que localize um novo provedor a fim de receber um novo fluxo de vídeo a partir do ponto solicitado (figura 8). Esta requisição é tratada de forma similar à requisição de reinício de exibição, exceto pelo fato de o buffer estar vazio e portanto o cliente não pode iniciar a exibição do filme imediatamente. Este fato não afeta a qualidade percebida pelo usuário, pois estes já estão acostumados com a latência de

dispositivos mecânicos como o videocassete e o DVD. Além disso, o MCD permite reduzir a latência através do mecanismo de remendos para encher o buffer mais rapidamente.

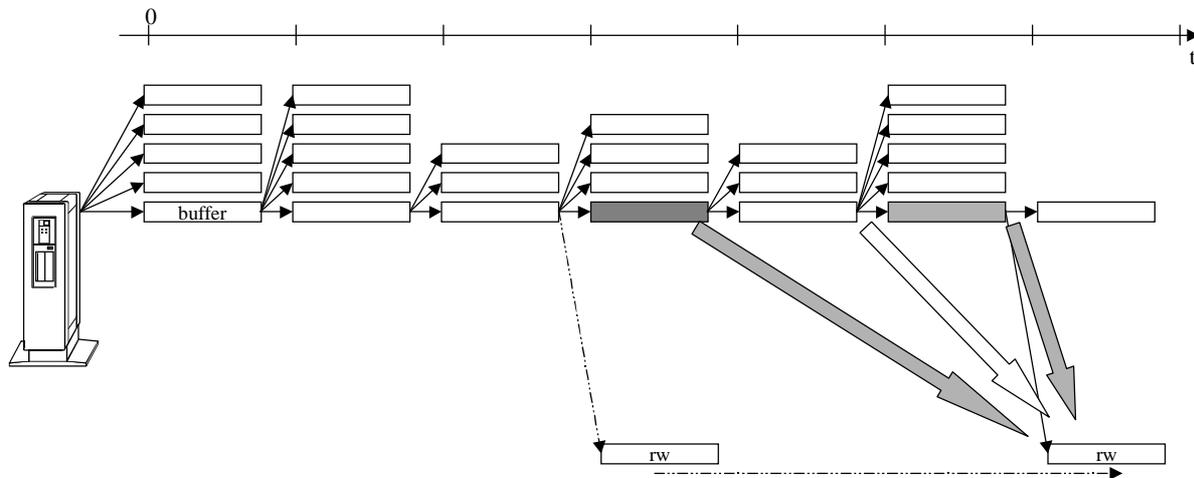


Figura 9: Retrocesso rápido contínuo na MCD

O segundo tipo de operações de ff/rw, o modo contínuo, funciona de forma similar ao modo discreto, exceto pelo fato de ao percorrer a árvore de buffers ir mandando o conteúdo dos buffers em rajada para o cliente que solicitou a operação. Como esta operação ocupa uma grande banda passante da rede, o cliente colaborador pode mandar apenas os quadros I ou quadros especiais com a média das imagens (figuras 9 e 10).

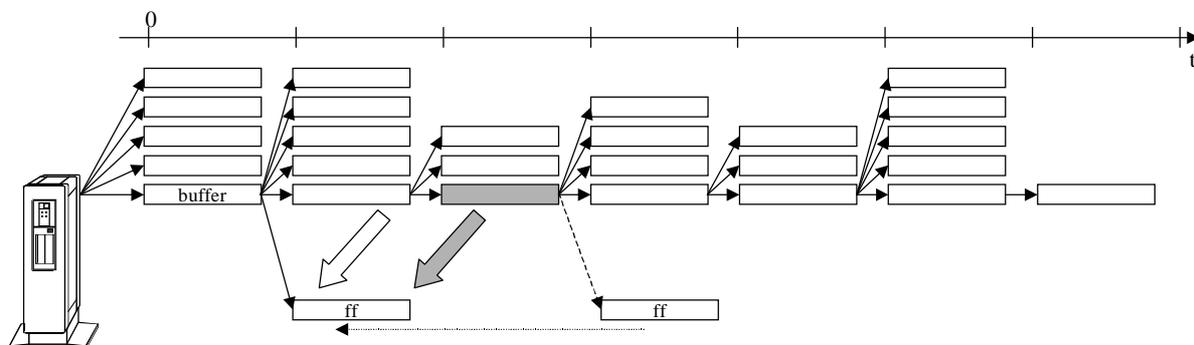


Figura 10: Avanço rápido contínuo na MCD

É importante observar que a MCD consegue oferecer estes serviços quando existem muitos clientes assistindo a um mesmo filme, caso o filme não seja popular a MCD não contribui para economizar os recursos do sistema, uma vez que as requisições terão que ser atendidas pelo servidor. Uma forma de educar o usuário com o objetivo de maximizar o uso da MCD é através da tarifação, com filmes populares tendo uma tarifa mais barata.

4 Estudo de performance

Nesta seção serão apresentados resultados preliminares de nossas simulações para mostrar a viabilidade desta nova abordagem, o sistema de VoD peer-to-peer com MCD. Primeiramente será descrito o ambiente de simulação.

O Ambiente de simulação

Para avaliar a performance da MCD, supôs-se que o servidor e os clientes estariam interconectados por uma rede ATM. Não se levou em consideração a topologia da rede, apenas que a capacidade da matriz de comutação do comutador ATM é capaz de atender todas as portas simultaneamente de tal forma que a banda passante interna do comutador possa ser explorada pela MCD. Esta consideração é análoga a considerar uma rede com topologia totalmente ligada, ou seja, a dorsal da rede não é o gargalo do sistema. Nesta fase do estudo considerou-se uma rede com serviço determinístico, onde toda a transmissão de informação respeita os atrasos contratados, sem atrasos e sem perdas devido a congestionamentos na rede. Posteriormente pretende-se investigar o comportamento da MCD com um serviço estatístico e em outras topologias.

Todos os clientes têm um buffer, suficiente para armazenar até 80 segundos de vídeo MPEG1. Considerando que um vídeo MPEG1 tem uma vazão média de 1.5 Mbps, um buffer de 16 MB é suficiente. Com as estações de trabalho atualmente comercializadas tem um mínimo de 64 MB, mais do que suficiente para suportar o sistema operacional Windows ou o Linux mais suas aplicações, usar um quarto dessa quantidade de memória é razoável. Para transmitir/receber este vídeo, o cliente precisa de uma vazão média de 1.5 Mbps full duplex, logo um enlace de 10 Mbps é um enlace superestimado para transmitir um fluxo multicast mais os comandos da MCD e os *patches*.

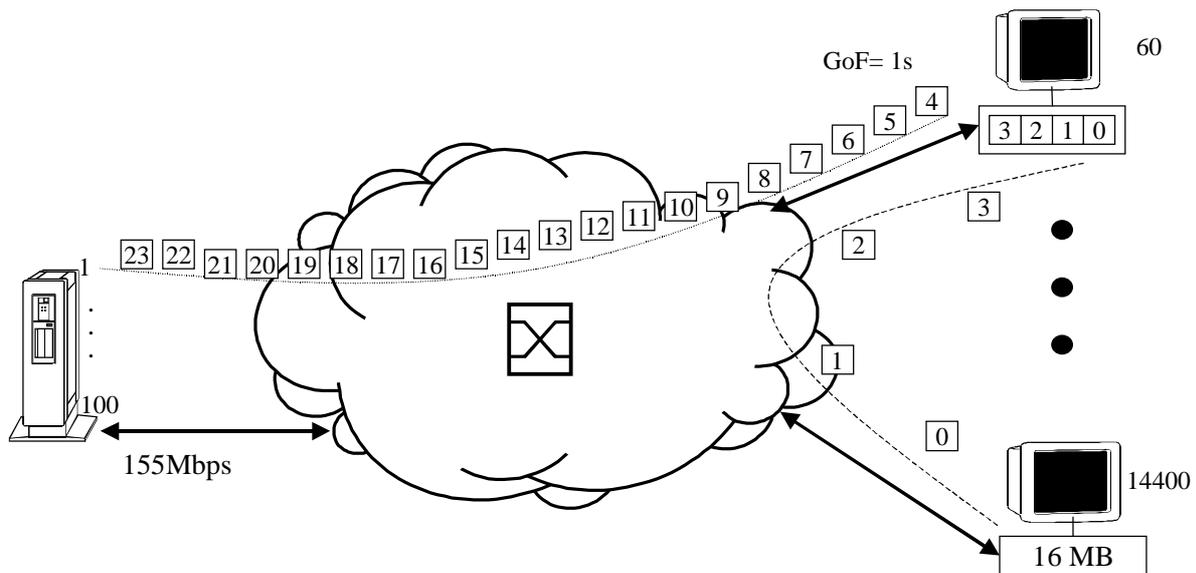


Figura 11: Ambiente da simulação

O servidor está conectado à rede ATM por um enlace de 155 Mbps. Com esta banda passante é possível suportar em média 100 canais lógicos. Um canal lógico é composto de todos os recursos necessários do servidor e seu enlace para sustentar um fluxo de vídeo com uma qualidade especificada. Um computador *off-the-shelf* e de baixo custo é capaz de prestar este serviço sem muito esforço. Dez por cento dos canais lógicos são reservados para atender pedidos de interatividade, o que diminui a possibilidade de haverem bloqueios durante a exibição de um filme.

Novos usuários chegam ao sistema segundo um processo de Poisson. O servidor contém um vídeo, pois o objetivo é estudar o comportamento da MCD. O próximo passo no

estudo consiste em verificar a quantidade de filmes que um servidor de VoD de baixo custo consegue suportar usando a MCD.

Quando um novo cliente chega ao sistema, o simulador cria um objeto cliente com um identificador. Este cliente é capaz de receber um fluxo de vídeo, armazená-lo em um buffer, mandá-lo para outros clientes através de um fluxo multicast ou um *patch* em rajada, e é claro, exibir o filme ao usuário.

Para simular a interatividade do cliente, simulou-se um cliente com probabilidade de 5% de fazer uma pausa em média a cada minuto e meio, com uma duração de 1 a 600 segundos uniformemente distribuído.

Todos estes parâmetros foram escolhidos para demonstrar a capacidade da MCD escalar um sistema de VoD com uma rede ATM de 155 Mbps.

A carga de trabalho e os parâmetros utilizados estão resumidos na tabela 1. Na coluna *default* estão os valores normalmente usados na simulação e na coluna **faixa** os valores que variaram durante a simulação.

As medidas de performance tomadas foram:

- Taxa de pedidos bloqueados – é a porcentagem das requisições para ver um novo filme que não puderam ser atendidas por falta de canais lógicos livres;
- Taxa de falha – é a porcentagem de clientes que experimentam falhas de exibição (glitches) por falta de GoFs no buffer;
- Vazão média – é a soma da média das vazões de cada fluxo multicast no sistema necessárias para atender a todos os usuários.
- Taxa de utilização do servidor – É a porcentagem de canais lógicos ocupados no servidor.

O simulador foi implementado em ANSI C++ e cada simulação foi executada por 7200 segundos, o que corresponde ao horário nobre das 20:00 às 22:00 horas. Todas as medidas tem um nível de confiança de 90% com um intervalo de confiança de $\pm 10\%$. As curvas foram interpoladas com o método de Bezier.

Parâmetros	default	faixa
Duração do vídeo (segundos)	3600	3600 e 7200
Tamanho médio do GoF (segundos)	1	-
Quantidade de canais lógicos (banda passante do servidor)	100	-
Tamanho do buffer (segundos)	80	
Taxa de requisições (chegadas/minuto)	-	0.5 a 60
Número de clientes	-	50 a 14400
Tempo médio entre interações (segundos)	90	-
Probabilidade de pausa	5%	-
Duração da pausa (segundos)	1-600	-

Tabela 1: Parâmetros da simulação

Vazão

A figura 12 mostra que o sistema convencional satura para uma taxa de requisição acima de 2 chegadas/min. Isto porque todos os canais lógicos disponíveis no servidor são rapidamente consumidos.

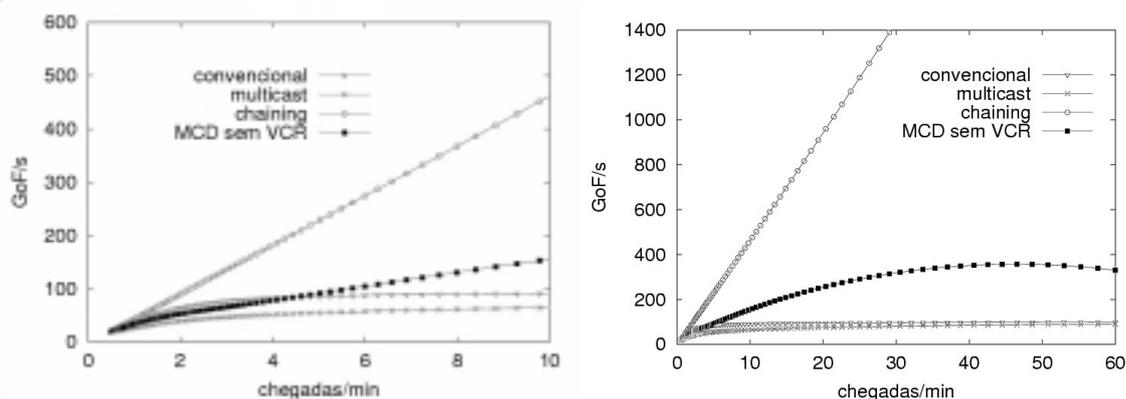


Figura 12: Vazão, o gráfico da esquerda tem uma escala maior para uma melhor visualização das curvas.

O sistema que usa Chaining tem uma vazão linear crescente com o aumento da taxa de requisição, mostrando que o sistema já não é mais limitado pelo número de canais lógicos oferecidos pelo servidor ou pelo enlace de acesso do servidor, entretanto ele precisa de uma banda passante crescente na dorsal da rede, o que limita a sua escalabilidade.

O sistema multicast usa menos banda passante no enlace de acesso do servidor porque consegue servir mais de um cliente com o mesmo fluxo de vídeo, no entanto a taxa de ocupação do servidor é alta comparada com o do sistema com MCD, como será mostrado mais adiante.

O sistema com MCD sem interação mostrou que com o crescimento da taxa de chegada a vazão cresce até 45 chegadas/min e após este ponto de máximo decresce. Isto ocorre porque após este ponto de máximo a MCD consegue explorar a grande massa de memória coletiva e fluxos multicast já estabelecidos, o que confirma o potencial da escalabilidade da MCD.

Taxa de utilização do servidor

Observando as curvas mostradas na figura 13 pode-se notar que tanto o sistema convencional como o multicast tem uma grande carga de trabalho, mesmo para um único filme no servidor. Este detalhe é muito importante, porque a duração do filme considerado era de 60 minutos, o que permite distribuí-lo em 100 canais, cada canal com um slot de 35 segundos. Mas se o filme tiver 120 minutos, 100 canais não serão suficientes para atender todas as chegadas, o que leva a uma taxa de bloqueio maior, como pode ser visto na figura 14.

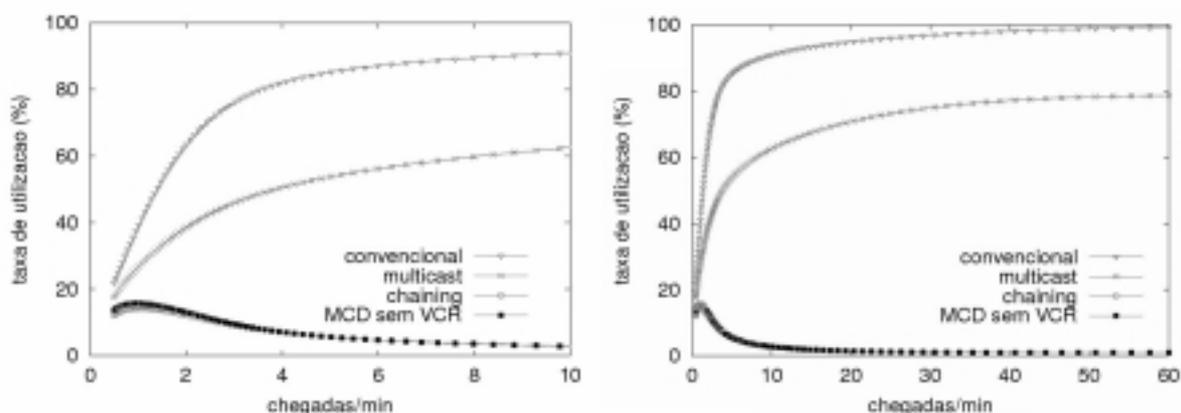


Figura 13: Taxa de utilização do servidor, o gráfico da esquerda tem uma escala maior para melhor visualização das curvas.

Por outro lado, o uso da MCD faz com que a taxa de utilização do servidor diminua com o aumento da taxa de chegadas, com uma performance quase igual a Chaining, mas usando menos banda passante da rede. A taxa de utilização do servidor cai a menos de 1% com uma taxa de chegada acima de 10 requisições/minuto. Ou seja, a MCD pode sustentar a escalabilidade do sistema com apenas um servidor de baixo custo disponibilizando muito mais do que um filme, ao contrário do sistema multicast que consegue servir apenas um filme de no máximo uma hora de duração.

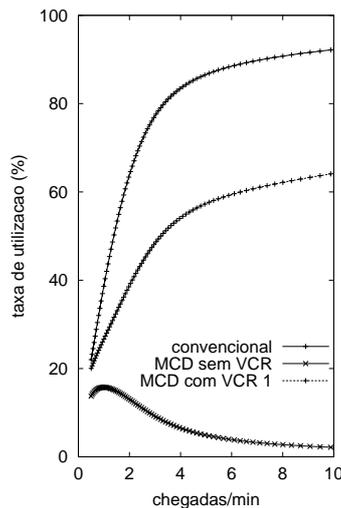


Figura 14: Taxa de utilização para um filme de 2 horas

Efeitos da interação da MCD

Com o objetivo de verificar o comportamento da MCD face às operações de VCR, foi implementada a operação de pausa/reinício no simulador [2]. Os algoritmos iniciais testados para achar um novo provedor de fluxo de vídeo a partir de qualquer ponto do filme são simples e foram implementados para dar uma idéia da viabilidade da MCD com a atividade de interação do usuário. Os algoritmos são baseados nas técnicas descritas anteriormente, a única diferença é a forma de escolher o novo provedor de um conjunto de possíveis candidatos. O primeiro algoritmo testado escolhe o primeiro da lista dos possíveis provedores. Como esta escolha obedece apenas o critério de menor esforço na escolha do novo provedor o resultado se mostrou ruim logo de início e por isso foi abandonado. Outro critério utilizado foi o de escolher entre os possíveis provedores aquele que causa a menor perturbação na cadeia já existente. Esta heurística é baseada no fato de que MCD sem interação tem uma boa performance, portanto tentar preservar a cadeia criada sem interação pode levar a bons resultados. De fato, conseguiu-se um bom resultado, mas ainda longe da performance ideal, a performance da MCD sem interação.

Para se ter uma idéia do comportamento da MCD com interação, comparou-se com os resultados da MCD sem interação e do sistema convencional, uma vez que o sistema convencional permite todas as operações de VCR.

A figura 15 mostra que a taxa de bloqueio da MCD com interação é baixa, entretanto a taxa de falhas aumenta a partir de 3 chegadas/minuto. Observa-se também que a MCD com interação tem uma taxa de utilização do servidor menor que a do sistema convencional, mas

muito maior do que a da MCD sem interação o que mostra a possibilidade de se extrair ganhos de performance com técnicas mais sofisticadas para gerenciar a MCD sob interatividade.

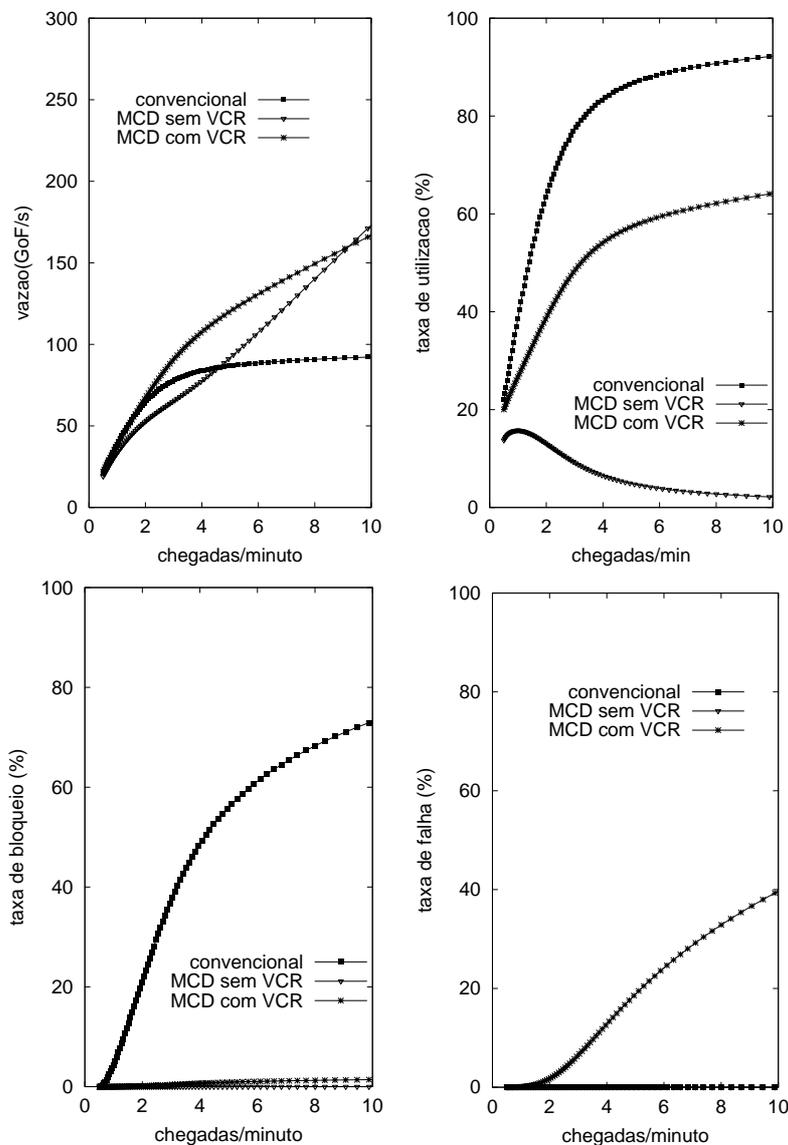


Figura 15: Gráficos da vazão, taxa de utilização, taxa de bloqueio e taxa de falha da MCD com interação

5 Trabalhos relacionados

O uso da memória do cliente para aumentar a escalabilidade de um sistema de VoD é novo. Além de Chaining [7] e Earthworm [12], que é uma extensão de Chaining, relativamente poucos trabalhos anteriores são relatados usando-se a memória do cliente para aumentar a performance de sistemas de tempo real. Entretanto, pode-se citar trabalhos que usam a memória do cliente para sistemas que não são em tempo real como o Napster [10] e o Gnutella [1] que usam a capacidade de armazenamento dos clientes para distribuir arquivos. No Napster existe um coordenador central que cataloga a localização de cada arquivo. O cliente que quer baixar um arquivo o localiza através do coordenador que tem um cadastro com os clientes

ativos que podem fornecer o arquivo, o provedor. O provedor Napster usa uma pequena parcela de sua banda passante para mandar o arquivo requisitado. Qualquer usuário que queira distribuir um arquivo MP3 precisa apenas de um cliente Napster e o arquivo MP3, é claro, que será registrado no coordenador. A abordagem do Gnutella é ligeiramente diferente do Napster porque ele não possui um coordenador central, sendo totalmente distribuído, além de poder compartilhar qualquer tipo de arquivo e não apenas arquivos MP3. Além disso, ambos os sistemas baixam todo o arquivo antes de sua exibição e em um sistema de VoD os arquivos são baixados de forma progressiva.

Também existem trabalhos que fazem uso da memória dos clientes para aumentar a performance de sistemas de arquivos [8] ou para aumentar o desempenho da paginação em uma memória virtual [4], no entanto estes estudos são para o uso de mídia contínua.

6 Considerações finais

A novidade da MCD está no uso do espaço de armazenamento do cliente (buffer) não só como uma linha de retardo, mas como um espaço de memória com uma visão única que pode ser gerenciada de forma cooperativa para oferecer funcionalidade de VCR a sistemas de VoD de forma escalável. O sistema de VoD resultante tem as seguintes vantagens:

- o usuário pode ver a exibição do filme escolhido sem grande espera, sem a necessidade de esperar o buffer encher ou aguardar o próximo slot de multicast;
- o usuário pode avançar e retroceder o filme continuamente e não apenas na parte armazenada no buffer do cliente, deixando o servidor livre de ter que reservar alguns canais lógicos só para prestar este serviço;
- ser escalável ao minimizar a carga do servidor quando o filme está armazenado na memória coletiva do sistema;
- ser mais tolerante a falhas, pois torna disponível várias cópias do filme na memória coletiva que pode ser usada em eventos inesperados, como uma falha em um enlace da rede, através do uso de um fluxo alternativo de vídeo da memória coletiva.

Os resultados preliminares adquiridos através de detalhada simulação mostraram que a MCD tem um grande potencial para aumentar a performance e a escalabilidade de sistemas de VoD. Sem interação, a MCD se mostrou melhor do que os esquemas propostos anteriormente. Com interação, apenas a MCD e o sistema convencional podem oferecer plena capacidade de operação VCR, sendo que a MCD tem melhores resultados. As técnicas pesquisadas até agora são muito simples, mesmo assim elas mostraram uma melhora na performance. Com a pesquisa de novas e melhores técnicas para gerenciar a MCD com interação, teremos resultados melhores. Simular a MCD com interação confirmou que ela pode ser usada para se implementar operações VCR, o que falta é investigar como fazê-lo sem perder a escalabilidade.

É importante ressaltar que a MCD permite o uso de servidores de VoD de baixo custo, tornando o serviço de VoD viável em grande escala usando poucos recursos. De fato, o que torna a MCD escalável é o número de clientes que usa o sistema. Quanto mais clientes, mais memória coletiva o que melhora a qualidade de serviço e a escalabilidade da MCD.

Bibliografia

- [1] Gnutella's Home Page. *http://gnutella.wego.com*.
- [2] K.C. Almeroth; M.H. Ammar. The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communication*, 14(6), 1996.
- [3] K.A. Hua; Y.Cai e S. Sheu. Patching: A multicast technique for true video-on-demand services. *ACM Multimedia '98*, 1998.
- [4] M. J. Feeley et all. Implementing global memory management in a workstation cluster. *Proceedings of the 15th ACM symposium on Operating Systems Principles*, december, 1995.
- [5] The ATM Forum Technical Committee. Audiovisual Multimedia Services: Video on Demand Specification 1.1. *af-saa-0049.001*, 1997.
- [6] ISO/IEC IS 13818-2 ITU-T Recommendation H.262. Information Technology – generic coding of moving pictures and associated audio – part 2: Video.
- [7] S.Sheu; K.A. Hua and W. Tavanapong. Chaining: A generalized batching technique for video-on-demand systems. *IEEE International Conference on Multimedia Computing and Systems*, june 1997.
- [8] T. Anderson; M. Dahlin, R. Wang e D. Patterson. Cooperative caching: using remote client memory to improve file system performance. *Proceedings of Usenix OSDI*, pages 267-280, november 1994.
- [9] L. Golubick; J.C.S. Lui e R.R. Muntz. Adaptative piggybacking: a novel technique for data sharing in video-on-demand storage servers. *Multimedia Systems*, 1996(4), pages 140-155.
- [10] Napster. Napster's Home Page. *http://www.napster.com*.
- [11] A. Dan; D. Sitaram e P. Shabuddin. Dynamic batching policies for an on-demand video server. *Multimedia Systems*, 1996(4), pages 112-121.
- [12] K.A. Hua, S. Sheu e JZ. Wang. Earthworm: A Network Memory Management Technique for Large-Scale Distributed Multimedia Applications. *IEEE INFOCOM '97*, Kobe, Japan.