

Novas Técnicas de Compartilhamento de Banda para Servidores de Vídeo sob Demanda com Interatividade *

Carlo Kleber da S. Rodrigues , Rosa Maria Meri Leão

¹ Universidade Federal do Rio de Janeiro,
COPPE/PESC
CxP 68511, Rio de Janeiro RJ 21941-972, Brazil

{kleber,rosam}@land.ufrj.br

Abstract. *This work presents two new bandwidth sharing techniques for video on demand servers with interactivity: Efficient Interactive Patching - PIE and Complete Interactive Patching - PIC. We still obtain an approximation for a threshold used for merging and opening data streams in the system, discuss over other related thresholds, and investigate the use of local buffer in the client side. Through simulations and competitive analysis we show for instance that the bandwidth savings are significant and that the approximation proposed for the threshold is adequate in certain scenarios. We are also able to quantify the influence of the local buffer on the bandwidth optimization and on the system complexity.*

Resumo. *Este trabalho apresenta duas novas técnicas de compartilhamento de banda para servidores de vídeo com interatividade: Patching Interativo Eficiente - PIE e Patching Interativo Completo - PIC. Obtemos ainda uma aproximação para um limiar de tempo usado em decisões de união e abertura de fluxos, discutimos sobre outros limiares de tempo relacionados, e investigamos a importância do uso de buffer local no cliente. Através de simulações e análises competitivas mostramos que a otimização de banda é bastante satisfatória e que a aproximação proposta para o limiar de tempo é adequada para certos cenários. Também quantificamos a influência do buffer local na otimização da banda e na complexidade do sistema.*

1. Introdução

Técnicas de compartilhamento de banda são indispensáveis para prover maior escalabilidade para sistemas de vídeo sob demanda (VoD) [14, 26]. Como principal contribuição deste trabalho propomos então duas novas técnicas de compartilhamento de banda admitindo o serviço com interatividade: *Patching* Interativo Eficiente - PIE e *Patching* Interativo Completo - PIC. Além disso, obtemos uma aproximação analítica para um limiar de tempo usado em decisões de união de fluxos e abertura de novos fluxos, e ainda discutimos sobre outros limiares de tempo relacionados. Também analisamos a importância da

*Este trabalho é parcialmente financiado pelo CNPq e Faperj.

utilização de *buffer* local no cliente para fins de evitar que unidades do objeto já transmitidas pelo servidor precisem ser requisitadas novamente pelo cliente. Por fim, avaliamos a complexidade do sistema.

Através de simulações e análises competitivas – em cenários baseados em servidores reais – com técnicas já apresentadas na literatura, mostramos por exemplo que a otimização de banda conseguida pelo emprego das novas técnicas é bastante satisfatória. Constatamos também que a aproximação do limiar de tempo obtida é bem próxima do valor ideal experimental para certos cenários, e que o uso de *buffer* local no cliente pode ter influência significativa na otimização da banda do servidor, na banda agregada dos clientes e na complexidade do sistema.

O restante deste trabalho está organizado conforme explicado a seguir. Inicialmente, na Seção 2, comentamos sobre os principais trabalhos relacionados. Em seguida, a Seção 3 introduz alguns conceitos preliminares e apresenta as novas técnicas de compartilhamento de banda aqui propostas. A derivação de um limiar de tempo usado nas decisões de abertura e união de fluxos e também a discussão sobre outros limiares de tempo relacionados estão na Seção 4. A Seção 5 é dedicada à apresentação dos mais importantes resultados de simulação obtidos nos diferentes cenários considerados. Por último, as conclusões e as propostas para trabalhos futuros constituem a Seção 6.

2. Trabalhos Relacionados

Banker *et al.* [18] em 1994 foram os primeiros a introduzir o conceito de *sintonia* nos servidores de VoD então existentes, os quais simplesmente realizavam a difusão (*broadcast*) de cópias do mesmo objeto em canais distintos e defasados no tempo entre si. A *sintonia* consistia em eficientemente determinar qual dos canais existentes estava transmitindo a unidade de dados desejada (ou mais próxima) pelo cliente. Um pouco mais tarde, houve as propostas de Dan *et al.* [13] e Almeroth e Ammar [3]. Dan *et al.* [13] tiveram especial atenção com as ações VCR (i.e., ações de interatividade praticadas pelo cliente) do tipo *Pause/Resume*, mas não discutiram claramente outros tipos de interatividade e a união de fluxos no sistema. Almeroth e Ammar [3] tiveram como principal idéia a utilização do *buffer* local do cliente para garantir um serviço contínuo.

Liao e Li [24] em 1997 propuseram uma técnica chamada *Split and Merge (SAM) protocol*. Esta proposta implementa todos os tipos de ações VCR e é a primeira a formalmente discutir a união de fluxos em andamento. A idéia básica é alocar um fluxo exclusivo de interação I tão logo o cliente execute uma ação VCR. Como característica principal desta técnica, tem-se o fato de que o *buffer* utilizado para prover a união de fluxos localiza-se nos nós de acesso da rede. Em seguida, Abram-Profeta e Shin [1] melhoram a proposta do *SAM protocol* com a idéia de utilizar um *buffer* local e exclusivo no cliente. Por volta de 1999, Poon e Lo [29] propuseram a técnica conhecida por *Single-Rate Multicast Double-Rate Unicast (SRMDRU)*. Esta técnica também admite todos os tipos de ações VCR. A idéia base é a utilização de fluxos exclusivos de interação I , como na proposta de Liao e Li [24], só que, após a ação VCR ter sido completada, o cliente passa a receber em uma taxa duas vezes maior que a taxa de exibição do objeto para o mais rapidamente possível alcançar um fluxo em andamento e então ser unido a ele.

Ma e Shin [25] em 2001 apresentaram a técnica conhecida como *Best-Effort*

Patching (BEP). Sua idéia básica segue a concepção da técnica original de acesso seqüencial *Patching* com a diferença de que a estrutura de união de fluxos é uma floresta de árvores de até três níveis de profundidade. Além desta proposta, existem outras pesquisas sobre modificações do esquema original de *Patching* para admitir florestas de árvores de três ou mais níveis (por exemplo: [7, 21, 32, 25]). Entretanto, os resultados sugerem que o nível de complexidade para a implementação e a gerência do sistema não justificam a sua implementação em face dos resultados observados de otimização de banda. Netto [27] e Gorza [20] apresentaram a técnica chamada *Patching Interativo*. A principal diferença desta técnica em relação à proposta de Ma e Shin [25] é que a estrutura de união de fluxos mantém-se em dois níveis (conforme a da técnica original *Patching*). Outra característica desta técnica é a possibilidade de atender uma requisição com um fluxo transmitindo uma unidade anterior à unidade solicitada.

Apesar de originalmente propostas para o acesso seqüencial, as técnicas baseadas em *união hierárquica de fluxos* (por exemplo, [15, 16, 17, 9, 22]), em geral podem ser modificadas ou servir de base para o desenvolvimento de técnicas para emprego em cenários com interatividade. No entanto, o custo de implementação e de gerência do sistema (complexidade) para estas técnicas pode ser bem maior que aqueles das técnicas baseadas em *Patching*, haja visto que a estrutura de união de fluxos passa a ser potencialmente uma floresta de árvores sem limites de profundidade e largura [4].

3. Técnicas Para Interatividade

3.1. Conceitos preliminares

Considere um servidor de vídeo e um grupo de clientes recebendo objetos multimídia (filmes, *video clips*, etc.) através da Internet. Os clientes sempre requisitam o início do objeto e assistem ao mesmo até o final sem interrupções. O cliente possui *buffer* local capaz de armazenar pelo menos metade do objeto requisitado e sua banda corresponde a duas vezes a taxa de exibição deste objeto. Por fim, os fluxos iniciados (*multicast* ou *unicast*) transmitem individualmente na mesma taxa de exibição do objeto requisitado.

A técnica *Patching* opera conforme explicado a seguir. O servidor escalona um fluxo *multicast* do objeto inteiro no momento da chegada do primeiro cliente. Os clientes seguintes, que requisitam o mesmo objeto e chegam dentro de um limiar de tempo, denominado *janela*, escutam o fluxo *multicast* (armazenando em *buffer* local para posterior exibição) e simultaneamente recebem a parte inicial do objeto, denominada *patch*, através de fluxos *unicast* individuais. Os clientes que chegam após o término da janela iniciam uma nova janela e o processo é reiniciado. Note que o *buffer* local do cliente é usado para fins de sincronismo, garantindo uma exibição contínua dos dados (unidades do objeto) recebidos a partir dos fluxos *multicast* e *unicast*.

Em [19, 6] foram propostos modelos semelhantes para estimativa da banda média para *Patching*. Nestes modelos foi suposto que as chegadas de requisições obedecem a um processo de Poisson de taxa λ_j para o objeto O_j . A banda média do servidor para transmissão do objeto O_j , medida em unidades da taxa de exibição do objeto, foi mostrada como sendo dada por $B_j = (T_j + \frac{\lambda_j W_j^2}{2}) / (W_j + \frac{1}{\lambda_j})$, onde T_j e W_j são a duração total e a janela do objeto O_j , respectivamente, e são expressos na mesma unidade de tempo.

Diferenciando-se a expressão de B_j em relação à W_j e igualando o resultado a zero, obtemos o valor ótimo da janela (Equação 1). E substituindo este resultado na própria expressão de B_j , obtemos a banda média do servidor para a técnica *Patching* quando é usada a janela ótima: $B_{j,otimizada} = \sqrt{2N_j + 1} - 1$, onde $N_j = \lambda_j T_j$ é o número médio de requisições que chegam durante o período de tempo T_j . N_j é comumente chamado de popularidade. Por simplicidade, denotamos $W_{j,otimo}$ e $B_{j,otimizada}$ por W_j e B_j , respectivamente.

$$W_{j,otimo} = \frac{\sqrt{2N_j + 1} - 1}{\lambda_j} \quad (1)$$

Em um cenário com interatividade temos as mesmas considerações do início desta subseção, adicionando apenas a característica do cliente possuir interatividade, ou seja, poder iniciar a exibição em qualquer posição do objeto e durante a exibição executar ações de pausa, retrocesso e avanço. O cenário básico descrito a seguir nos ajuda a entender a forma atual mais comum de atendimento das requisições para um objeto armazenado no servidor.

Assuma que o objeto de duração T está dividido em unidades de mesmo tamanho: u_1, \dots, u_T . Uma requisição ocorre em $t = t_0$ e existem n fluxos *multicast* em andamento. Seja $u_{desejada}$ a unidade do objeto solicitada por esta requisição e, sem perda de generalidade, admita que todos os fluxos S_i , $i = 1, \dots, n$, estão transmitindo unidades de dados posteriores à $u_{desejada}$. Por fim, seja $dist_i$ a distância entre $u_{desejada}$ e a unidade sendo atualmente transmitida pelo fluxo *multicast* S_i . O atendimento da solicitação de $u_{desejada}$ pode ser realizado de duas formas: serviço *descontínuo* ou serviço *contínuo* [26, 1].

No caso de serviço *descontínuo*, a solução consiste em simplesmente identificar o fluxo que tenha associado o menor valor de $dist_i$. Seja então S_{min} este fluxo e $dist_{min}$ o valor de distância associado. Note que se $dist_{min} > 0$ então o cliente que fez a solicitação por $u_{desejada}$ receberá na realidade uma unidade posterior àquela solicitada e distante $dist_{min}$, caracterizando o serviço *descontínuo* por não estar recebendo exatamente aquilo que foi solicitado. No caso do serviço *contínuo*, o valor $dist_{min}$ é determinado e então é verificado se ele está ou não dentro de um limiar de tempo permitido, aqui denominado Delta Posterior. Caso esteja, o cliente é atendido pelo fluxo correspondente S_{min} e as unidades do objeto que faltam (relativas à distância $dist_{min}$) são enviadas através de um canal *unicast*. Caso $dist_{min}$ seja superior ao Delta Posterior, um novo fluxo *multicast* $S_{n+1} = S_{novo}$ é então aberto para atender a nova requisição. Atualmente a determinação do Delta Posterior é feita de forma empírica. Posteriormente obtemos uma expressão para obtenção deste parâmetro.

Uma opção para evitar a criação de um fluxo S_{novo} é a união do cliente com um outro fluxo que esteja transmitindo uma unidade de dados anterior mas suficientemente próxima à $u_{desejada}$. A estimativa de quão suficientemente próxima é bastante subjetiva e está atrelada ao grau de tolerância do cliente. Seja Delta Anterior o limiar de tempo utilizado para mensurar esta proximidade, e seja $S_{anterior}$ o fluxo identificado nesta situação. Estimulando a utilização do Delta Anterior, Almeroth e Ammar [3] inclusive sugerem que clientes de sistemas interativos podem não se importar com pequenas discontinuidades em relação ao serviço oferecido.

3.2. Novos critérios para união de fluxos

Na subseção anterior apresentamos critérios que usualmente são utilizados para realizar a união de fluxos em cenários com interatividade. A seguir, apresentamos três novos critérios que usamos parcial ou integralmente nas propostas de PIE e PIC.

Para entendimento do primeiro critério, considere o seguinte cenário. Um novo fluxo S_{novo} é criado para atender a requisição da unidade $u_{desejada}$ feita por um cliente c . Admita a existência de um outro fluxo em andamento, denominado S_{perto} , que esteja suficientemente próximo de S_{novo} e não tenha clientes realizando *patch*, de tal sorte que seja eficiente, em termos de otimização de banda do servidor, unir estes dois fluxos, isto é, tornar S_{novo} alvo de S_{perto} . A escuta simultânea desses dois fluxos tem sua duração determinada pela diferença inicial entre as unidades sendo transmitidas por S_{perto} e S_{novo} , respectivamente, que denotamos por Delta Perto. Note que nenhuma solicitação adicional em termos de unidades de dados é feita ao servidor para que esta operação possa ser conduzida. Ao final do tempo denotado por Delta Perto, os clientes de S_{perto} têm a mesma quantidade de dados armazenada que os clientes de S_{novo} e, portanto, podem passar a escutar exclusivamente o fluxo S_{novo} . O fluxo S_{perto} é então extinto. Note que é necessário estabelecer um valor máximo para Delta Perto tal que essa união seja de fato eficiente.

Em relação ao segundo critério, admita o cenário em que um fluxo recém iniciado S_{novo} , então escalonado como alvo do fluxo já em andamento no sistema S_{perto} , é extinto por não ter mais clientes associados antes que o próprio S_{perto} consiga alcançá-lo. Nesta situação, a união de S_{perto} e S_{novo} não mais acontecerá. Os clientes de S_{perto} passam portanto a serem clientes *órfãos*. Para contornar esta situação, no momento da extinção do fluxo S_{novo} , o servidor realiza uma busca no sistema para identificar um outro fluxo em andamento que possa ser o substituto de S_{novo} . Seja $S_{substituto}$ este fluxo. Assim, a economia de banda passa a ocorrer em função da união de S_{perto} com $S_{substituto}$. Os clientes de S_{perto} deixam então de ser órfãos. Caso $S_{substituto}$ seja extinto antes que S_{perto} consiga alcançá-lo, um novo procedimento de busca por um substituto para $S_{substituto}$ é realizado recursivamente. O limiar de tempo considerado para identificação do fluxo $S_{substituto}$ é também feito igual a Delta Perto.

Por último, o terceiro critério diz respeito ao que chamamos de *ação espontânea* do servidor para realizar a união de fluxos *multicast* em andamento. A idéia é localizar um fluxo em andamento no sistema $S_{destino}$ que possa ser o alvo de um fluxo S_{origem} que acabou de ter todos os *patches* associados terminados, ou seja, que a partir deste instante pode tentar se unir a um fluxo que esteja transmitindo uma unidade de dados posterior a do mesmo. Naturalmente também é necessário estabelecermos um limiar de tempo para que essa união seja eficiente. Aqui denotamos este limiar por Delta Destino.

3.3. Técnica Patching Interativo Eficiente – PIE

Nesta técnica fazemos uso apenas do primeiro critério definido na subseção anterior e a premissa básica é a manutenção de uma estrutura de união de fluxos de mesma complexidade que a técnica *Patching*. O Algoritmo 1 traz a operação simplificada desta técnica quando do recebimento de uma requisição para unidade $u_{desejada}$. Note que a primeira pesquisa por fluxo (*linha 2*) ocorre considerando o conceito de Delta Anterior pois esta é a menos onerosa por evitar que qualquer unidade do objeto tenha de ser transmitida. A pesquisa seguinte utiliza o conceito de Delta Posterior (*linha 4*), e neste caso é necessário

utilizar um *patch* para suprir a parte inicialmente perdida (*linha 5*). Caso não sejam identificados fluxos capazes de atender a requisição é então aberto um novo fluxo (*linha 6*) e verificado se existe algum outro fluxo que pode ser unido a ele no limite de Delta Perto (*linha 7*). Como observação final, é importante dizer que a união de clientes com fluxos em andamento (*linha 5*) e a união de fluxos (*linha 7*) não necessariamente se dão com sucesso. Isto porque, devido à aleatoriedade do comportamento dos clientes, os fluxos podem ser extintos antes de serem alcançados e o próprio cliente também pode desistir de alcançar um determinado fluxo.

Algoritmo 1: Patching Interativo Eficiente

```

1 Receber requisição por  $u_{desejada}$ ;
2 se ( $\exists S_{anterior}$ ) então
3    $join(u_{desejada}, S_{anterior})$ ;
   senão
4   se ( $\exists S_{posterior}$ ) então
5      $patch(u_{desejada}, S_{posterior})$ ;
   senão
6      $start(u_{desejada}, S_{novo})$ ;
7     se ( $\exists S_{perto}$ ) então  $merge(S_{perto}, S_{novo})$ ;
   fim
fim

```

3.4. Técnica Patching Interativo Completo – PIC

As técnicas PIE e PIC são bem semelhantes. O que difere uma da outra é que para PIC usamos uma extensão do primeiro critério e também os segundo e terceiro critérios (definidos na Subseção 3.2) conforme sucintamente relembramos a seguir. Quando um novo fluxo S_{novo} é criado, busca-se a identificação de um conjunto de fluxos do tipo S_{perto} , em vez de apenas um fluxo como em PIE, caracterizando uma extensão do primeiro critério. No insucesso da união de dois fluxos *multicast* devido ao término precoce do fluxo a ser alcançado (fluxo alvo), existe a imediata tentativa de se buscar no sistema algum outro fluxo em andamento (no limite de Delta Perto) que possa substituir este, impedindo assim, quando possível, que os clientes tornem-se permanentemente órfãos. Isto corresponde ao segundo critério. E o terceiro critério é a tentativa de união de fluxos em andamento. Ela é iniciada no momento em que um fluxo *multicast* qualquer deixa de ter *patches* associados, dado que este fluxo não tenha ainda um fluxo alvo a ser alcançado.

4. Parâmetros Delta

As seguintes suposições são consideradas no modelo para obtenção de uma aproximação para Delta Posterior: (i) as chegadas de requisições são representadas por um processo de Poisson; (ii) ao dividirmos o objeto em um número inteiro de unidades, temos que a probabilidade de acesso a cada uma destas unidades é exatamente a mesma; (iii) o padrão de acesso é tal que cada chegada de requisição requisita sempre um tamanho médio L do objeto; e (iv) apenas as requisições que solicitam a mesma unidade do objeto podem compartilhar o fluxo inicialmente disparado por uma delas. Com estas suposições, é possível

então mostrar que uma aproximação para Delta Posterior pode ser dada por D_{otimo} , apresentado na Equação 2. (Detalhes do modelo estão em [11].)

$$D_{otimo} = \frac{\sqrt{2\frac{\beta}{T}L+1}-1}{\frac{\beta}{T}}, \quad (2)$$

onde β é a taxa do processo de Poisson que representa as chegadas de requisições por novas unidades, L e T denotam o tamanho médio solicitado pelas requisições e o tamanho do objeto armazenado no servidor, respectivamente, medidos em número de unidades. O valor de D_{otimo} é tão mais preciso quanto maior a proximidade entre o modelo analítico e o modelo da carga real. Esta proximidade pode ser medida a partir da localidade de acesso às unidades do objeto e do desvio padrão de L . Caso esta proximidade seja julgada insatisfatória, T deve ser feito igual ao número de unidades diferentes solicitadas no início da requisição. Esta redefinição empírica de T reflete a tentativa de adequação do modelo analítico, que admite uma distribuição de acesso uniforme, à verdadeira distribuição de acesso presente na carga real.

A obtenção analítica de Delta Anterior, Delta Perto e Delta Destino não é trivial de ser realizada devido a complexidade do modelo de interatividade do cliente. Alternativamente, procedemos então como indicado a seguir. Fixamos Delta Anterior em 10 segundos por ser um valor aparentemente bem tolerável para a maioria das aplicações de VoD. Os parâmetros Delta Perto e Delta Destino assemelham-se por estabelecerem a distância máxima permitida entre fluxos *multicast* para que os mesmos possam ser unidos. Daí, assumimos Delta Perto sempre igual a Delta Destino e testamos experimentalmente diversos valores na tentativa da identificação de valores ideais.

5. Avaliação de Performance

5.1. Objetivos e Métricas

Considerando diferentes cenários de interatividade, os experimentos têm os seguintes objetivos: (i) analisar a distribuição da banda do servidor; (ii) quantificar a importância da utilização, por parte do cliente, de *buffer* local de tamanho igual ao do objeto para fins de armazenamento permanente das unidades do objeto transmitidas pelo servidor e conseqüentemente evitar que unidades já transmitidas necessitem ser requisitadas novamente; (iii) determinar valores *ideais* para os parâmetros Delta Posterior, Delta Perto e Delta Destino; (iv) analisar o número de clientes ativos (i.e., clientes realizando *streaming*); (v) realizar análises competitivas das novas técnicas com a técnica *Patching* Interativo [27, 20] e com uma versão modificada para acesso não-sequencial que implementamos a partir da técnica *Closest Target* [16].

Em relação ao objetivo (ii) cabe ainda acrescentar o seguinte. As técnicas PIE e PIC, assim como também *Patching* Interativo, *Closest Target* e *Patching* fazem uso em sua concepção de um *buffer* local no cliente de tamanho igual a pelo menos metade do tamanho do objeto, sendo este usado estritamente para fins de sincronismo da escuta, permitindo assim exibição contínua das unidades do objeto (Seção 3.1). Ressaltamos que, com a utilização da idéia de *buffer* de tamanho do objeto e armazenamento de todas as unidades recebidas, a função de sincronismo continua existindo, o que ocorre é a adição da possibilidade de armazenar permanentemente as unidades do objeto com fins de evitar,

como já explicado, que unidades já armazenadas tenham de ser novamente requisitadas ao servidor. Por simplicidade de referência e clareza de discussão, nos experimentos comparativos utilizamos então o sufixo SB (*Simple Buffer*) para indicar que a técnica faz uso apenas do *buffer* simples convencional, de tamanho igual a pelo menos metade do objeto e com fins exclusivos de sincronismo, e o sufixo CB (*Complete Buffer*) para indicar que a técnica relacionada faz uso de *buffer* completo de mesmo tamanho do objeto e utilizado para prover sincronismo e armazenar permanentemente todas as unidades do objeto enviadas pelo servidor.

Escolhemos as técnicas *Patching* Interativo e *Closest Target* para fins de comparação competitiva com nossas propostas pelos seguintes motivos. A técnica *Patching* Interativo, doravante referenciada por PI, é bem recente e tem como uma das principais premissas a estrutura de união de fluxos como uma floresta de árvores de no máximo dois níveis, garantindo significativa simplicidade de gerência do sistema. Para compararmos PIE (ou PIC) com PI, empregamos então a conceituação de *razão competitiva* [5, 8] diretamente nas distribuições de banda obtidas. Mais precisamente, definimos as razões competitivas $RM_{max} = \max\left(\frac{P[Banda > k]_{PI}}{P[Banda > k]_{PIE \text{ ou } PIC}}\right)$ e $RM_{min} = \min\left(\frac{P[Banda > k]_{PI}}{P[Banda > k]_{PIC \text{ ou } PIE}}\right)$. Esclarecemos que estas razões consideram ou PI e PIE (ou PIC) com o uso de *buffer* simples ou PI e PIE (ou PIC) com a utilização de *buffer* completo e, devido à precisão dos resultados de simulação, apenas valores de $P[Banda > k] \geq 10^{-4}$. Com isto desprezamos as caudas das distribuições. Já a técnica *Closest Target*, por ser baseada em união hierárquica de fluxos, tem como principal atrativo a sua eficiência em potencial. Esta eficiência esperada se dá pela característica de pesquisa exaustiva e irrestrita por fluxos no intuito de compartilhamento de dados, e também pelo fato de todos os fluxos abertos serem *multicast* e daí potencialmente fluxos alvos. Neste caso a discussão se volta, não somente para a comparação da otimização de banda conseguida, mas também para a quantificação do nível de complexidade introduzido no sistema. Avaliamos esta complexidade pelo cálculo do trabalho realizado pelo servidor como explicado no parágrafo seguinte. A versão modificada para acesso não-sequencial de *Closest Target* que implementamos é doravante referenciada simplesmente por CT.

O trabalho realizado pelo servidor é proporcional ao número de mensagens recebidas e às operações realizadas na ocorrência das mensagens. A operação de maior custo é a busca por um fluxo no sistema. Fazemos a análise de pior caso e admitimos a busca por um fluxo para as técnicas PIC, PIE e PI sendo dada por $O(n)$ e para CT sendo dada por $O(m)$, onde n e m representam o número de fluxos *multicast* abertos no sistema devido ao emprego da técnica. Temos que $n = \frac{m}{FA}$, onde FA é um fator de aglutinação de fluxos proporcional ao valor de Delta Posterior. Notadamente a determinação analítica de m e FA (e conseqüentemente de n) não é simples de ser realizada. De forma alternativa, obtemos os valores de m e n experimentalmente através da análise das curvas de banda do próprio servidor. Com estes valores e o valor das taxas de mensagens também obtidas experimentalmente (quais sejam: TB – taxa relacionada a mensagens de pedidos de unidades do objeto, TM – taxa de mensagens para término de união de fluxos, TP – taxa de mensagens para término de *patch*, e TF – taxa de mensagens para saída do cliente do sistema), quantificamos o trabalho realizado pelo servidor. A Tabela 1(a) sumariza as complexidades de tempo devido ao recebimento de cada tipo de mensagem a depender da técnica utilizada, onde $O(C)$ corresponde a uma complexidade de tempo constante. Os detalhes destas complexidades podem ser encontrados em [12]. Podemos então quan-

tificar o trabalho médio para as técnicas PI, PIE, PIC e CT por meio das formulações apresentadas na Tabela 1(b).

Técnica	msg TB	msg TM	msg TP	msg TF
PIC	$O(2n)$	$O(n)$	$O(n)$	$O(n)$
PIE	$O(n)$	$O(C)$	$O(C)$	$O(C)$
PI	$O(n)$	—	$O(C)$	$O(C)$
CT	$O(2m)$	$O(m)$	—	$O(m)$

(a) Complexidades

Técnica	Formulação
PI	$TB * n + TP + TF$
PIE	$TB * 2n + TM + TP + TF$
PIC	$TB * 2n + (TM + TP + TF) * n$
CT	$TB * 2m + (TM + TF) * m$

(b) Formulações

Tabela 1: (a) Complexidade para cada tipo de mensagem e (b) Formulações para o trabalho médio realizado.

5.2. Caracterização dos Cenários

Em nossos experimentos utilizamos cargas sintéticas provenientes dos servidores de ensino à distância eTeach [30, 2, 10] e MANIC [28, 23], e do provedor Universo *Online* [31, 10], ou simplesmente UOL, um dos maiores provedores de conteúdo da América Latina. As principais características destas cargas estão apresentadas na Tabela 2(a). Examinamos três diferentes cenários no total, onde cada cenário refere-se a carga de um destes servidores. Para efeito de análise, consideramos o atendimento de um único objeto constituído de unidades (indivisíveis) de 1 seg cada. As ações VCR consideradas são: *Play*, *Stop*, *Pause/Resume*, *Jump Forwards* e *Jump Backwards*.

Estadística	eTeach	UOL	MANIC
tam. obj. T (seg)	2199	226	4175
popularidade obj. N	98	97	99
total num. requisições	5146	1114	677
num. requisições por sessão	10.29	2.23	1.35
tam. médio por requisição L (seg)	118	134	1190
desvio padrão de L (seg)	143	91	1184
coef. de variação de L	1.21	0.68	1.00
num. unid. acessadas (de início)	≈ 2199	≈ 24	≈ 24

(a) Estatísticas

eTeach			UOL		
Técnica	RMin	RMax	Técnica	RMin	RMax
PIE-SB	1.0	1.8	PIE-SB	0.7	2.0
PIE-CB	0.6	1.6	PIE-CB	0.9	14.5
PIC-SB	1.0	2.0	PIC-SB	0.7	2.0
PIC-CB	1.0	2.5	PIC-CB	0.9	14.5

MANIC		
Técnica	RMin	RMax
PIE-SB	1.0	16.3
PIE-CB	1.0	25.8
PIC-SB	1.0	15.1
PIC-CB	1.0	14.1

(b) Razões Competitivas

Tabela 2: (a) Estatísticas das cargas sintéticas e (b) Razões competitivas para PIE e PIC com relação à PI.

5.3. Descrição dos Experimentos

Salvo mencionado diferentemente, fizemos Delta Perto e Delta Destino iguais a Delta Posterior, e fixamos Delta Anterior em 10 segundos. Devido às restrições de espaço, não ilustramos e detalhamos todos os experimentos referenciados neste texto. Resultados e figuras completas, além de outros experimentos pertinentes, podem ser encontrados em [12, 11].

A seguir descrevemos os experimentos realizados em cada um dos cenários: (i) para Delta Posterior igual a 50% do valor da janela ótima de *Patching* W (Equação 1),

comparamos a distribuição da banda do servidor para as técnicas PIE, PIC e PI, e ainda calculamos as razões competitivas relacionadas. Estimativas diferentes de 50% de W para Delta Posterior redundam qualitativamente nos mesmos resultados comparativos. Este experimento nos fornece condições de analisar competitivamente as técnicas mencionadas; (ii) para Delta Posterior igual a 50% do valor da janela ótima de *Patching* e considerando a técnica PIE ou PIC (a de melhor performance), comparamos as distribuições da banda do servidor, como também as curvas de clientes ativos (i.e., clientes realizando *streaming*), com *buffer* simples (SB) e com *buffer* completo (CB). A curva de clientes ativos aqui chamamos de banda agregada dos clientes. Estimativas diferentes de 50% de W para Delta Posterior redundam qualitativamente nos mesmos resultados comparativos. Este experimento nos dá condições então de quantificar a importância do emprego de *buffer* completo através da otimização conseguida na banda do servidor e na banda agregada dos clientes; (iii) para Delta Posterior variando aproximadamente no intervalo de 0 – 300% do valor do tamanho médio solicitado por requisição L , observamos a distribuição da banda do servidor obtida para PIE ou PIC (a de melhor performance). Este experimento nos dá condições de determinar o valor ideal de Delta Posterior e avaliar a acurácia da aproximação analítica que fizemos (Equação 2). Valores fora do intervalo mencionado se mostram ineficientes para fins de otimização; mais precisamente, o valor médio L tende a ser o limite superior do intervalo pois corresponde ao tamanho médio de fluxo a ser compartilhado; (iv) para Delta Perto variando aproximadamente no intervalo de 0 – 300% do valor ideal de Delta Posterior, observamos a distribuição da banda do servidor obtida para PIE ou PIC (a de melhor performance). Este experimento nos dá condições de determinar o valor ideal de Delta Perto. Valores fora do intervalo mencionado se mostram ineficientes para fins de otimização. Lembramos que sempre fazemos Delta Destino igual ao valor de Delta Perto, e Delta Anterior fixo e igual a 10 seg; (v) considerando valores ideais dos parâmetros Delta, comparamos a distribuição da banda do servidor para as técnicas PIE ou PIC (a de melhor performance) e CT, e ainda calculamos o trabalho médio realizado pelo servidor. Este experimento permite análises competitivas das técnicas mencionadas.

5.4. Síntese e Análise dos Resultados

Para o Cenário 1 – Carga eTeach – mencionamos o seguinte. As técnicas PIE e PIC têm melhor desempenho que a técnica PI (Tabela 2(b)). PIE e PIC têm desempenhos semelhantes com tênue vantagem para PIC. O uso de *buffer* local completo traz significativa otimização da banda do servidor (Figuras 1(a) e 1(b)) e da banda agregada dos clientes (Figura 1(c)), redução de 31.08% no pico). O valor ideal experimental de Delta Posterior é 30-40% da janela ótima e sua aproximação analítica (Equação 2) é satisfatória (39.89% da janela ótima). O valor ideal experimental obtido para Delta Perto é o valor ideal experimental de Delta Posterior. CT tem melhor desempenho que PIC (Figuras 2(a) e 2(b)) à custa de maior complexidade (Tabela 3 e Figura 1(d)).

Para o Cenário 2 – Carga UOL – temos o seguinte. As técnicas PIE e PIC têm melhor desempenho que a técnica PI (Tabela 2(b)). PIE e PIC têm desempenhos bem semelhantes, com vantagem para PIE pela sua maior simplicidade de implementação. O uso de *buffer* local completo traz significativa otimização da banda do servidor e da banda agregada dos clientes (redução de 18.85% no pico). O valor ideal experimental de Delta Posterior é 100-110% da janela ótima e sua aproximação analítica (Equação 2) não é satisfatória (244% da janela ótima). O valor ideal experimental obtido para Delta Perto

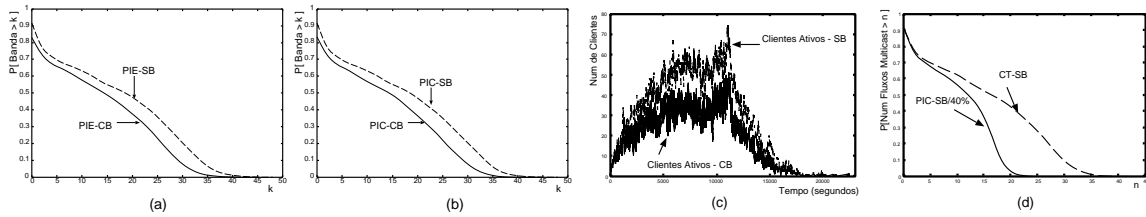


Figura 1: Cenário 1 – (a) Distribuição da banda para PIE; (b) Distribuição da banda para PIC; (c) Clientes ativos; (d) Distribuição de fluxos *multicast*.

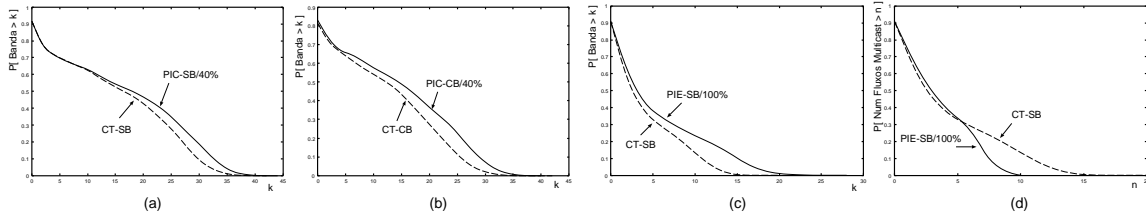


Figura 2: Cenário 1 – (a) Distribuição da banda-SB (PIC e CT) (b) Distribuição da banda-CB (PIC e CT); Cenário 2 – (c) Distribuição da banda-SB (PIE e CT); (d) Distribuição de fluxos *multicast*-SB (PIE e CT).

é o valor ideal experimental de Delta Posterior. CT tem melhor desempenho que PIE (Figura 2(c)) à custa de maior complexidade (Tabela 3 e Figura 2(d)). De forma interessante, a utilização de *buffer* completo neste cenário faz com que o número máximo de fluxos *multicast* da técnica CT aumente (Tabela 3). Isto porque o uso de *buffer* completo leva à redução do número de fluxos *multicast* e esta redução pode diminuir a probabilidade de compartilhamento e a união de fluxos, provocando extraordinariamente um aumento momentâneo do número de fluxos em relação àquele do cenário com *buffer* simples.

Já no Cenário 3 – Carga MANIC – temos o seguinte. As técnicas PIE e PIC têm melhor desempenho que a técnica PI (Tabela 2(b)). PIE e PIC têm desempenhos semelhantes, com vantagem para PIE pela sua maior simplicidade de implementação. O uso de *buffer* local completo traz otimização discreta da banda do servidor e da banda agregada dos clientes (redução de 2.22% no pico). O valor ideal experimental de Delta Posterior é 70-80% da janela ótima e sua aproximação analítica (Equação 2) não é satisfatória (146% da janela ótima). O valor ideal experimental obtido para Delta Perto é o valor ideal experimental de Delta Posterior. CT tem melhor desempenho que PIE (Figura 3(a)) à custa de maior complexidade (Tabela 3 e Figura 3(b)).

A partir dos resultados observados nos três cenários descritos temos então as

Parâmetro	eTeach		UOL		MANIC	
	PIC	CT	PIE	CT	PIE	CT
max{num fluxos multicast} (SB)	24	42	9	19	11	17
trabalho realizado (SB)	7176.0	12841.9	367.2	2962.1	22.1	159.0
max{num fluxos multicast} (CB)	22	42	9	22	11	16
trabalho realizado (CB)	9658.0	29875.4	191.9	2155.6	20.9	129.9

Tabela 3: Avaliação de complexidade.

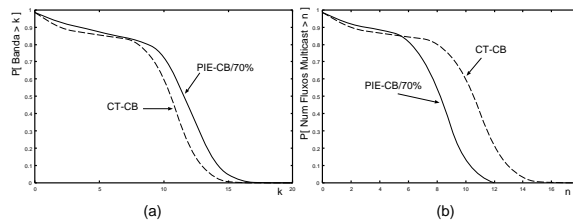


Figura 3: Cenário 3 – Distribuições (a) da banda e (b) de fluxos *multicast*.

seguintes constatações globais. Primeiro, as técnicas PIE e PIC apresentam níveis de otimização de banda bastante semelhantes e satisfatórios, o que termina favorecendo PIE para aplicação prática devido a sua maior simplicidade de implementação. Segundo, PIE e PIC têm melhor desempenho – quantificada pelas razões competitivas e distribuição da banda do servidor – que PI. Terceiro, a técnica CT apresenta melhor desempenho que PIE e PIC à custa de uma maior complexidade do sistema – aproximadamente uma ordem de grandeza – inferida a partir do cálculo do trabalho médio realizado pelo servidor e que se traduz principalmente pelo maior número de mensagens transmitidas, processamento mais complexo, e maior dependência na implementação de roteamento *multicast*. Quarto, os resultados advindos da investigação da importância do uso de *buffer* no cliente indicam que os benefícios trazidos dependem do cenário em questão e se refletem potencialmente na banda do servidor, na banda agregada dos clientes, e na complexidade do sistema. Quinto e último, para emprego prático de PIE e PIC, podemos admitir: a aproximação analítica obtida para Delta Posterior é satisfatória em cenários de interatividade média/alta (i.e., num. requisições por sessão > 10); em cenários de interatividade baixa, devemos utilizar como estimativa inicial para Delta Posterior ou o valor do tamanho médio da requisição L ou o valor da janela ótima W da técnica *Patching* (pois os valores ideais de Delta Posterior estiveram sempre confinados a percentuais de L e W), e a partir daí determinarmos dinamicamente o valor ideal; Delta Perto e Delta Destino devem ser iguais ao valor de Delta Posterior; para Delta Anterior, fixamos 10 seg por ser um valor aparentemente tolerável para a maioria das aplicações de serviço de VoD.

6. Conclusões e Trabalhos Futuros

Este trabalho apresentou duas novas técnicas de compartilhamento de banda para servidores de vídeo com interatividade: *Patching Interativo Eficiente* - PIE e *Patching Interativo Completo* - PIC. Obtivemos ainda uma aproximação para um limiar de tempo (Delta Posterior) usado em decisões de união e abertura de fluxos, discutimos sobre outros limiares de tempo relacionados (Delta Anterior, Delta Perto e Delta Destino) e, por fim, investigamos a importância do uso de *buffer* local do cliente para evitar que dados já armazenados sejam novamente requisitados ao servidor. Através de simulações e análises competitivas mostramos principalmente que a otimização de banda devido ao emprego das novas técnicas é bastante satisfatória e superior ao da recente proposta PI [27, 20]. Também observamos que CT (Seção 5.1) tem melhor performance que PIE e PIC à custa de uma maior complexidade do sistema. Ainda constatamos que a aproximação analítica para Delta Posterior é satisfatória para cenários de interatividade média/alta, e mostramos como estimar de forma prática os demais limiares de tempo definidos para PIE e PIC. Vimos ainda que o uso de *buffer* no cliente pode influenciar significativamente na com-

plexidade do sistema e também na otimização da banda do servidor e na banda agregada dos clientes. Como trabalhos futuros e em andamento temos: obter fórmulas analíticas para Delta Perto e Delta Destino; avaliar o impacto de Delta Anterior ≥ 10 seg na banda do servidor e na percepção do cliente; estudar o uso de janela ótima para técnicas de interatividade baseadas em união hierárquica de fluxos.

Por último, agradecemos à professora J. M. Almeida e ao doutorando M. Rocha (UFMG) por cederem as cargas sintéticas aqui utilizadas.

Referências

- [1] E. L. Abram-Profeta and K. G. Shin. Providing unrestricted VCR functions in multicast video-on-demand servers. In *Proc. of IEEE ICMCS*, pages 66–75, Austin, Texas, June 1998.
- [2] J. M. Almeida, J. Krueger, D. L. Eager, and M. K. Vernon. Analysis of educational media server workloads. In *Proc. 11th Int'l Workshop Network and Operating Systems Support for Digital Audio and Video (NOSSDAV'01)*, pages 21–30, June 2001.
- [3] K. C. Almeroth and M. H. Ammar. The use of multicast delivery to provide a scalable and interactive video-on-demand service. *IEEE Journal on Selected Areas in Communications*, 14(5):1110–1122, August 1996.
- [4] A. Bar-Noy, G. Goshi, R. E. Ladner, and K. Tam. Comparison of stream merging algorithms for media-on-demand. In *Proc. of MMCN 2002*, pages 18–25, San Jose, CA, January 2002.
- [5] A. Borodin and R. El-Yaniv. *On-line computation and competitive analysis*. Cambridge University Press, The Pitt Building, Trumpington Street, Cambridge, United Kingdom, 1998.
- [6] Y. Cai, K. Hua, and K. Vu. Optimizing Patching Performance. In *Proc. SPIE/ACM Conference on Multimedia Computing and Networking*, pages 204–215, 1999.
- [7] Y. Cai and K. A. Hua. An efficient bandwidth-sharing technique for true video on demand systems. In *Proc. of the 7th ACM Int'l Multimedia Conference (ACM Multimedia'99)*, pages 211–214, November 1999.
- [8] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. Competitive analysis of on-line stream merging algorithms. In *Proc. of the 27th Int'l Symposium on Mathematical Foundations of Computer Science (MFCS)*, pages 188–200, 2002.
- [9] Wun-Tat Chan, Tak-Wah Lam, Hing-Fung Ting, and Wai-Ha Wong. On-line stream merging, max span, and min coverage. In *Proc. of the 5th Conference on Algorithms and Complexity (CIAC)*, pages 337–346, 2003.
- [10] C. Costa, I. Cunha, A. Borges, C. Ramos, M. Rocha, J. M. Almeida, and B. Ribeiro-Neto. Analyzing client interactivity in streaming media. In *Proc. 13th ACM Int'l World Wide Web Conference*, pages 534–543, May 2004.
- [11] C. K. da S. Rodrigues. Mecanismos para compartilhamento de recursos para transmissão de mídia contínua na internet. Exame de Qualificação, UFRJ, COPPE/PESC, Julho 2004.
- [12] C. K. da S. Rodrigues and R. M. M. Leão. Novas técnicas de compartilhamento de banda para servidores de vídeo sob demanda com interatividade. Relatório Técnico, UFRJ-COPPE/PESC, Dezembro 2004.
- [13] A. Dan, D. Sitaram, P. Shahabuddin, and D. Towsley. Channel allocation under batching and VCR control in movie-on-demand servers. *Journal of Parallel and Distributed Computing*, 30(2):168–179, November 1995.

- [14] E. de Souza e Silva, R. M. M. Leão, B. Ribeiro-Neto, and S. Campos. Performance issues of multimedia applications. *Lecture Notes in Computer Science*, 2459:374–404, July 2002.
- [15] D. Eager, M. Vernon, and J. Zahorjan. Minimizing bandwidth requirements for on-demand data delivery. In *Proc. 5th Int'l Workshop on Multimedia Information Systems (MIS'99)*, pages 80–87, October 1999.
- [16] D. Eager, M. Vernon, and J. Zahorjan. Optimal and efficient merging schedules for video-on-demand servers. In *Proc. 7th ACM Int'l Multimedia Conference (ACM Multimedia'99)*, pages 199–202, November 1999.
- [17] D. Eager, M. Vernon, and J. Zahorjan. Bandwidth skimming: A technique for cost-effective video-on-demand. In *Proc. IS&T/SPIE MMCN'00*, pages 206–215, January 2000.
- [18] R. O. Banker et al. Method of providing video-on-demand with VCR-like functions. U. S. Patent 5357276, 1994.
- [19] L. Gao and D. Towsley. Supplying instantaneous video-on-demand services using controlled multicast. In *Proc. IEEE Multimedia Computing Systems'99*, pages 117–121, June 1999.
- [20] M. L. Gorza. Uma técnica de compartilhamento de recursos para transmissão de vídeo com alta interatividade e experimentos. Tese de Mestrado, UFF, Departamento de Ciência da Computação, Dezembro 2003.
- [21] D. Guan and S. Yu. A two-level patching scheme for video-on-demand delivery. *IEEE Transactions on Broadcasting*, 50(1):11–15, March 2004.
- [22] E. G. Coffman Jr., P. Jelenkovic, and P. Momcilovic. Provably efficient stream merging. In *Proc. 6th Int'l Workshop on Web caching and Content Distribution*, Boston, MA, June 2001.
- [23] J. F. Kurose, H. I. Lee, J. Steinberg, and M. Stern. Manic: Multimedia asynchronous networked individualized courseware. Technical Report 96-72, Department of Computer Science, University of Massachusetts, Oct 1996.
- [24] W. Liao and V. O. K. Li. The split and merge protocol for interactive video-on-demand. *IEEE Multimedia*, 4(4):51–62, Oct 1997.
- [25] Huadong Ma and K. G. Shin. A new scheduling scheme for multicast true VoD service. *Lecture Notes in Computer Science*, 2195:708–715, 2001.
- [26] Huadong Ma and K. G. Shin. Multicast video-on-demand services. *ACM SIGCOMM Computer Communication Review*, 32(1):31–43, 2002.
- [27] B. C. M. Netto. Patching interativo: Um novo método de compartilhamento de recursos para transmissão de vídeo com alta interatividade. Tese de Mestrado, UFRJ, COPPE/PESC, Fevereiro 2004.
- [28] J. Padhye and J. Kurose. An empirical study of client interactions with continuous-media courseware server. In *Proc. IEEE NOSSDAV*, Cambridge, UK, July 1998.
- [29] W. W. F. Poon and K. T. Lo. Design of multicast delivery for providing VCR functionality in interactive video-on-demand systems. *IEEE Transactions on Broadcasting*, 45(1):141–148, March 1999.
- [30] *eTeach - Learning on Demand*. <http://eteach.engr.wisc.edu/newEteach/home.html>.
- [31] *Universo Online*. <http://www.uol.com.br>.
- [32] Y. W. Wong and Jack Y. B. Lee. Recursive Patching - An efficient technique for multicast video streaming. In *Proc. 5th International Conference on Enterprise Information Systems (ICEIS) 2003*, pages 23–26, Angers, France, April 2003.