

# BIVoD: Um novo protocolo P2P para acesso interativo em sistemas de vídeo sob demanda

Luiz J. H. Filho<sup>1</sup>, Carlo K. da S. Rodrigues<sup>2</sup> e Rosa M. M. Leão<sup>1</sup>

<sup>1</sup>Programa de Engenharia e Sistemas de Computação – COPPE/PESC  
Universidade Federal do Rio de Janeiro – UFRJ  
CEP 21941-972 – CxP 68511 – Rio de Janeiro – RJ – Brasil

<sup>2</sup>Faculdade de Tecnologia e Ciências Sociais Aplicadas – FATECS  
Centro Universitário de Brasília – UniCEUB  
SEPN 707/907 - CEP 70790-075 – Brasília – DF – Brasil

{ljhfilho,rosam}@land.ufrj.br, {carlokleber}@gmail.com

**Resumo.** *Este artigo propõe um novo protocolo para sistemas de vídeo sob demanda com interatividade. Sua concepção é baseada no paradigma BitTorrent e tem, como inovações principais, a implementação de prioridades para os blocos de vídeo a serem recuperados e o emprego de um modelo de previsão de comportamento do usuário. A análise e a validação são feitas por meio de simulações usando cargas de um servidor multimídia real. Além disso, comparamos o protocolo proposto com outros da literatura. Os resultados finais evidenciam a eficiência do novo protocolo, apontando otimizações de até uma ordem de grandeza em algumas métricas de desempenho consideradas.*

**Abstract.** *This article presents a new protocol for interactive video-on-demand systems. Its conception is based on the BitTorrent paradigm. As main innovative aspects, we have that the video chunks are categorized and a user-behavior predicting model is used. The analysis and the validation are both carried out through simulations using workloads from a real multimedia server. Furthermore, we compare the new protocol with others of the literature. The final results outline the efficiency of this new protocol, showing optimizations of up to one order of magnitude over some competitive metrics used herein.*

## 1. Introdução

A arquitetura *peer-to-peer* (P2P) tem se destacado como solução para sistemas de vídeo sob demanda com interatividade (*trick mode VoD*). Nessa arquitetura, os clientes, denominados *peers* ou nós, colaboram com o servidor principal na distribuição do vídeo pela rede da qual fazem parte. Assim, tem-se o aumento da escalabilidade do sistema.

Uma das dificuldades de implementação desse tipo de arquitetura decorre do fato de não haver como garantir conexões estáveis e confiáveis entre os *peers*, pois os mesmos são livres para entrar e sair do sistema tão frequentemente quanto desejarem. Além disso, também é preciso lembrar que os protocolos utilizados para a distribuição de vídeo precisam levar em consideração algumas características da aplicação em si. No caso de vídeo sob demanda (VoD), o cliente pode realizar ações interativas como: saltos para frente e/ou para trás e pausas. Esse tipo de aplicação é também sensível ao atraso e tolerante apenas

a poucas perdas de pacotes, sob pena de comprometer a qualidade de serviço final. Visando resolver essas dificuldades e restrições, várias propostas já foram apresentadas na literatura como, por exemplo, a de Hoffmann et al [Hoffmann Filho et al. 2009].

Basicamente, existem três estruturas para a distribuição de vídeo P2P: a estrutura em árvore (*tree*), em malha (*mesh*) e a híbrida. Na estrutura em árvore, os *peers* se organizam em uma única ou em múltiplas árvores, nas quais a fonte de vídeo é a raiz da árvore. São formadas relações hierárquicas de pai e filho entre os nós, nas quais somente o pai encaminha o fluxo de vídeo a seus filhos. Sendo assim, cada *peer* que desejar participar da distribuição, deverá entrar na árvore e, sem que este faça requisições, começará a receber o fluxo de vídeo de seu pai. O trabalho apresentado em [Chu et al. 2000] é exemplo de proposta para a distribuição de vídeo que utiliza a estrutura em árvore.

Na estrutura em malha, não existe uma estrutura hierárquica definida, ou seja, os *peers* podem tanto enviar quanto receber blocos de qualquer outro *peer*. Um *peer* pode também receber dados de vários *peers* fontes simultaneamente. Os trabalhos apresentados em [Hoffmann Filho et al. 2009] e [Shah and Pâris 2007] fazem uso da estrutura em malha para a distribuição de vídeo. Por fim, a combinação dessa estrutura com a estrutura em árvore constitui a estrutura híbrida. Apesar de ser uma proposta que tenta unir o que há de melhor das duas estruturas, sua implementação e sua manutenção são de relativa complexidade. Por exemplo, o trabalho apresentado em [Li et al. 2007] propõe estruturas híbridas para a distribuição de vídeo.

Dentre os protocolos P2P da literatura, destaca-se o protocolo *BitTorrent* [Cohen, B. 2003] pela sua comprovada eficiência na replicação de arquivos. Por essa razão, recentemente, esse protocolo passou a ser considerado como uma promissora solução também para a distribuição de vídeo na Internet [Hoffmann Filho et al. 2009, Vlavianos et al. 2006]. De forma sucinta, duas políticas constituem a sua essência: a *política de seleção de vizinhos* e a *política seleção de blocos* do objeto. A primeira destina-se a selecionar os *peers* que podem servir de fonte de informação (*vizinhos*) para um outro *peer* que deseja pertencer a rede, para fins de receber um objeto (vídeo). Um *peer* considerado fonte de informação é um *peer* que possui os blocos do objeto de interesse. Essa seleção é feita de forma aleatória considerando um conjunto de *peers* informado por uma entidade central da rede denominada *tracker*. Por sua vez, a segunda política estabelece o critério a ser utilizado quando da escolha do bloco do objeto. Esse critério é o da raridade: o bloco que ocorre em menor número na rede é dito o mais raro. Quanto mais raro é o bloco, maior é a probabilidade dele ser solicitado aos vizinhos.

O desafio para a adaptação do protocolo *BitTorrent* para aplicações de VoD reside nos três pontos comentados a seguir. O primeiro é que existe a necessidade de que os blocos sejam recebidos dentro de um limite de tempo para que possam ser visualizados pelos clientes. Mas, como os blocos do objeto não são solicitados em sequência (mas em função da raridade), não é possível ter essa garantia. O segundo é que o protocolo *BitTorrent* faz uso de políticas de incentivo, privilegiando clientes que já estão na rede e que já participaram em algum momento da distribuição de blocos. Se por um lado isso evita clientes *egoístas* (i.e., clientes que desejam apenas receber dados), por outro lado, pode introduzir latência de serviço, fazendo com que clientes novos precisem esperar. A depender dessa latência, é possível que alguns clientes venham a abandonar o sistema sem terem sido servidos. Por último, o protocolo *BitTorrent* não prevê a interatividade do

usuário.

À luz do exposto, este artigo possui duas importantes contribuições. Primeiro, a apresentação de um novo protocolo P2P para sistemas interativos de VoD, denominado *BitTorrent Interactive Video on Demand* (BIVoD), o qual utiliza uma arquitetura P2P em malha, por sua simplicidade e baixa sobrecarga de controle para a manutenção da estrutura, e é baseado no protocolo *BitTorrent*. Segundo, a realização de uma extensa e detalhada análise competitiva entre algumas das mais recentes propostas de adaptações do protocolo *BitTorrent* para o serviço de VoD, e o novo protocolo aqui apresentado.

A avaliação e a validação do protocolo BIVoD são feitas por meio de simulações, usando cargas geradas a partir de *logs* reais do sistema multimídia RIO (*Random I/O System*) [Netto et al. 2005]. Diferentes métricas de performance são consideradas. Os resultados dos experimentos aqui realizados indicam otimizações de até uma ordem de grandeza em algumas das métricas, comparativamente às outras propostas da literatura.

O restante deste texto tem a organização descrita a seguir. A Seção 2 encerra o protocolo *BitTorrent* e o modelo de comportamento de usuários proposto em [De Vielmond et al. 2007], o qual é utilizado no novo protocolo BIVoD. A Seção 3 trata sobre os mais recentes trabalhos da literatura direcionados para a adaptação do protocolo *BitTorrent* para o serviço de VoD. Descreve-se detalhadamente o novo protocolo BIVoD na Seção 4. Na Seção 5, são apresentados e analisados os resultados obtidos nos experimentos. Por fim, as conclusões finais e os trabalhos futuros aparecem na Seção 6.

## 2. Fundamentos

### 2.1. Protocolo BitTorrent

Como descrito na seção anterior, o protocolo *BitTorrent* é essencialmente constituído pela *política de seleção de vizinhos* e pela *política de seleção de blocos*. A seguir são apresentados alguns conceitos adicionais [Carlsson and Eager 2007].

Os objetos a serem distribuídos pela rede são divididos em blocos (*pieces*), os quais, por sua vez, são divididos em sub-blocos (*sub-pieces*). Os sub-blocos de um mesmo bloco podem ser recuperados em paralelo a partir de diferentes nós (*peers*). Um *peer* é dito possuidor de um bloco quando todos os seus sub-blocos já foram recuperados. Um *peer* é dito interessado em todos os *peers* que possuem pelo menos um bloco que ele próprio não possui. Nesse protocolo, faz-se a diferenciação entre *peers* que possuem todo o objeto (*seeds*), e os *peers* que ainda não possuem todo o objeto (*leechers*). *Seeds* apenas transmitem, e *leechers* transmitem e recebem.

Um enxame (*swarm*) é um conjunto de *peers* que participam da transmissão e recepção de um mesmo objeto. Cada enxame é controlado por uma entidade central denominado *tracker*. Para pertencer a um enxame, é necessário que o *peer* contacte o *tracker*. Este, por sua vez, passa ao *peer* uma lista de outros *peers* que têm o objeto desejado. O *peer* então seleciona aleatoriamente um subconjunto da lista e inicia requisições para estabelecer conexões TCP bidirecionais. Se as conexões forem estabelecidas, esses *peers* do subconjunto passam a ser denominados *vizinhos* daquele *peer*.

O protocolo *BitTorrent* também utiliza uma política para determinar quais *peers* têm preferência para receber os blocos existentes no *swarm*. Essa política é vista como uma estratégia de incentivo, denominada *tit-for-tat*, e serve para evitar o *free-riding*, i.e.,

a condição de que *peers* ajam de forma egoísta e utilizem o enxame apenas para receber blocos, sem transmitir para nenhum outro *peer*. Sob essa estratégia, cada *peer* transmite tipicamente para os  $k$  *peers* que recentemente lhe permitiram as melhores taxas de recepção, muito embora ele possa receber solicitações de mais de  $k$  *peers* interessados em receber blocos do objeto que ele dispõe.

A recusa para transmitir para certos *peers* do enxame é chamada de bloqueio (*choking*). Também existe o processo de desbloqueio otimista (*optimistic unchoking*), que permite aos *peers* reservarem parte de sua banda para realizar transmissões para *peers* selecionados aleatoriamente, independentemente de sua contribuição ao enxame. Os processos de bloqueio e desbloqueio são realizados em intervalos regulares de tempo.

## 2.2. Modelo de Vielmond-Leão-Silva

A proposta feita em [De Vielmond et al. 2007] consiste em um modelo de Markov Oculto (HMM) hierárquico para emular o comportamento de usuários acessando um servidor multimídia. Esse modelo baseou-se em uma aplicação real de ensino a distância, onde alunos do curso do CEDERJ [de Souza e Silva et al. 2006] assistem a vídeo-aulas, previamente gravadas e sincronizadas com transparências que ficam armazenadas no sistema multimídia RIO [Netto et al. 2005]. A estrutura hierárquica desse modelo possui propriedades interessantes: a complexidade da fase de treinamento é menor que o HMM convencional, as dependências de curto prazo são capturadas pela cadeia da hierarquia inferior, e a dinâmica de longo prazo é governada pela cadeia de Markov oculta (hierarquia superior). A cadeia de Markov oculta governa a dinâmica de uma sessão de usuário, e os estados ocultos capturam a dependência das ações do usuário dentro do contexto de uma transparência. Assim, dentro de um estado oculto temos a dinâmica das ações do usuário.

## 3. Trabalhos Relacionados

### 3.1. Protocolos BiToS e de Zhou-Chiu-Lui

O protocolo *BitTorrent Streaming* (BiToS) [Vlavianos et al. 2006] é voltado para exibição sequencial do vídeo, i.e., sem interatividade. Ele é semelhante ao protocolo *BitTorrent*, excetuando-se apenas a sua *política de seleção de blocos*, conforme explicado a seguir.

Os blocos de vídeo a serem solicitados aos *peers* vizinhos são, para efeito de definição de prioridade, classificados em dois conjuntos:  $A_p$  e  $B_p$ . Os blocos do conjunto  $A_p$  são aqueles que ainda não foram recuperados e que estão próximos de serem visualizados (i.e., blocos de alta prioridade). Esse conjunto tem um tamanho fixo de  $m$  blocos. Quando um bloco não foi recuperado e já passou o tempo para que seja visualizado pelo usuário, diz-se que o bloco foi perdido. Os blocos do conjunto  $B_p$  são aqueles que ainda não foram recuperados e cujo instante de reprodução não está próximo.

A solicitação de um bloco do conjunto  $A_p$  é feita com probabilidade  $p$ , enquanto que a solicitação por um bloco do conjunto  $B_p$  é feita com probabilidade  $(1 - p)$ . A probabilidade  $p$  busca estabelecer um equilíbrio entre o que precisa ser visualizado imediatamente e a aquisição de blocos futuros. Essa probabilidade pode ser ajustada dinamicamente em função das condições de operação do sistema. O bloco a ser recuperado do conjunto  $A_p$  ou  $B_p$  é sempre o bloco mais raro, ou seja, aquele que estiver menos replicado na rede. Se um bloco é removido do conjunto  $A_p$ , então o próximo bloco da sequência

do conjunto  $B_p$  é transferido para  $A_p$ . Esse procedimento mantém a cardinalidade do conjunto  $A_p$  igual a  $m$ .

A proposta de Zhou-Chiu-Lui [Zhou et al. 2007], também voltada para a exibição sequencial do vídeo, é bem semelhante à proposta BiToS, que acabou de ser explicada. A diferença é que, para o conjunto de blocos de alta prioridade  $A_p$ , o bloco é recuperado considerando-se uma política sequencial.

### 3.2. Protocolo de Shah-Pâris

O protocolo de Shah-Pâris [Shah and Pâris 2007], também baseado no protocolo *BitTorrent*, distribui vídeo sob demanda sequencialmente, ou seja, não há interatividade. No entanto, tanto a *política de seleção de blocos* quanto a *política de seleção de vizinhos* do protocolo *BitTorrent* são alteradas, conforme explicado a seguir.

O vídeo é dividido em  $T$  blocos. É definida então uma janela deslizante  $J_d$  que encerra os  $w$  blocos em sequência que presumidamente têm a maior prioridade. Apenas os blocos contidos na janela  $J_d$  podem ser solicitados, e esta solicitação ocorre em função da raridade do bloco. Existem duas situações em que a janela desliza. Na primeira situação, a janela  $J_d$  desliza sempre que seu primeiro bloco for recuperado. Ela desliza até que a sua primeira posição seja referente a um bloco que ainda não foi recuperado. Na segunda situação, define-se um intervalo de tempo denominado *playback delay*. Decorrido esse intervalo de tempo, a janela  $J_d$  desliza por uma extensão de  $w$  blocos.

A *política de seleção de vizinhos* faz com que, a cada janela de  $w$  blocos recuperados, o nó selecione aleatoriamente  $n$  vizinhos da lista recebida do *tracker* para serem desbloqueados (*unchoking*). Nos demais instantes de tempo, é utilizado o próprio algoritmo original de *seleção de vizinhos* do protocolo *BitTorrent* para esse desbloqueio.

### 3.3. Protocolo de Hoffmann-Rodrigues-Leão

A proposta de Hoffmann-Rodrigues-Leão [Hoffmann Filho et al. 2009] é voltada para o acesso interativo, ou seja, o cliente pode executar ações de interatividade enquanto visualiza o objeto. Essa proposta difere do protocolo *BitTorrent* no que se refere à *política de seleção de blocos*, conforme explicado a seguir.

Os blocos do vídeo são classificados em dois conjuntos: *janela de playback* e  $B_p$ . O conjunto *janela de playback* contém os blocos de alta prioridade, ou seja, aqueles que presumidamente serão mais rapidamente acessados pelo usuário. Esse conjunto encerra  $m$  blocos consecutivos. O conjunto  $B_p$  possui os blocos de baixa prioridade, aqueles que presumidamente irão demorar mais até serem visualizados. Esse conjunto tem tamanho  $T - m$ , onde  $T$  é o tamanho total do vídeo em número de blocos. Dentro desse conjunto, é definido um subconjunto denominado *janela de previsão*. Esse subconjunto possui  $k$  blocos consecutivos, definidos a partir do modelo de comportamento de usuário apresentado em [De Vielmond et al. 2007].

Além de classificar os blocos de acordo com a prioridade, são utilizadas probabilidades para selecionar de qual conjunto recuperar os blocos. Com probabilidade  $p$ , escolhe-se a *janela de playback*. Com probabilidade  $(1 - p)$ , escolhe-se o conjunto  $B_p$ . Se escolhido o conjunto  $B_p$ , com probabilidade  $q$  são recuperados blocos de dentro da *janela de previsão*, e com probabilidade  $(1 - q)$  são recuperados blocos fora dela. A Figura 1(a) exemplifica essa descrição.

Os blocos da *janela de previsão* e os blocos do conjunto  $B_p$  são ambos selecionados em função da raridade. Para a seleção dos blocos da *janela de playback*, são definidas duas variantes: *Previsão Mais Raro* e *Previsão Sequencial*. Como as próprias denominações sugerem, a variante *Previsão Mais Raro* utiliza a política do bloco mais raro, enquanto que a variante *Previsão Sequencial* utiliza uma política sequencial. Na ocorrência de um salto do ponto corrente de visualização, a *janela de playback* é instantaneamente atualizada. Essa atualização faz com que a primeira posição dessa janela passe a se referir ao bloco para o qual o cliente saltou. Naturalmente, nessa situação, também são atualizados o conjunto  $B_p$  e a *janela de previsão*.

## 4. Um novo protocolo: BitTorrent Interactive Video on Demand

### 4.1. Motivação

Os experimentos realizados em [Hoffmann Filho et al. 2009] mostram que a proposta de Hoffmann-Rodrigues-Leão apresenta desempenho superior ao das propostas Bi-Tos [Vlavianos et al. 2006] e de Zhou-Chiu-Lui [Zhou et al. 2007], mas, em algumas situações, desempenho inferior ao da proposta de Shah-Pâris [Shah and Pâris 2007]. Isso deve-se ao mencionado a seguir.

Primeiro, a recuperação de blocos do conjunto de baixa prioridade faz com que blocos que não são necessários no instante imediato, nem em um futuro próximo, sejam recuperados em detrimento de outros que deveriam ser visualizados em um curto intervalo de tempo. Daí, ocorre a falta de blocos. Segundo, no início da visualização de um vídeo ou na ocorrência de um salto, ocorrem faltas de blocos consecutivos, provocando o efeito *toca e pára*. Por último, o fato de deslocar a *janela de playback* para o bloco inicial do salto (mesmo que o salto seja para dentro da então *janela de playback*) aumenta o número de blocos a serem recuperados na janela, pois inicia-se uma nova *janela de playback*. Daí, aumenta-se a probabilidade de ocorrer falta de bloco.

Do exposto, teve-se então a motivação de desenvolver o protocolo *BitTorrent Interactive Video on Demand* (BIVoD) na expectativa de suplantar os problemas identificados.

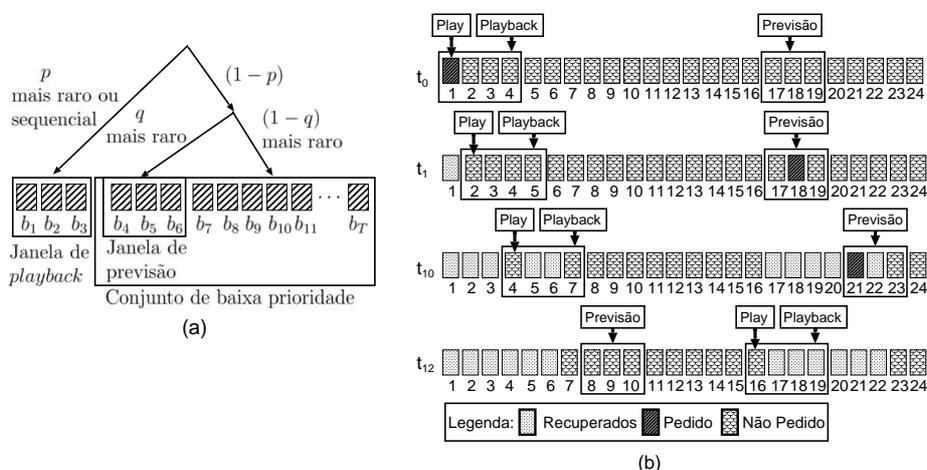


Figura 1. (a) Proposta de Hoffmann-Rodrigues-Leão; (b) Proposta BIVoD.

## 4.2. Descrição do novo protocolo

A proposta BIVoD prevê que o cliente possa realizar ações de interatividade enquanto assiste ao vídeo. Essa proposta difere do protocolo *BitTorrent* no que se refere à *política de seleção de blocos*.

Os blocos do vídeo são classificados em dois conjuntos: *janela de playback* e *janela de previsão*. A escolha de qual janela selecionar o bloco, da *janela de playback* ou da *janela de previsão*, ocorre de forma alternada. O conjunto *janela de playback* contém os blocos de alta prioridade, ou seja, aqueles que supostamente serão mais rapidamente acessados pelo usuário. Esse conjunto possui  $m$  blocos consecutivos, os quais são selecionados em função do critério da raridade.

Por sua vez, o conjunto *janela de previsão* contém blocos que possivelmente serão selecionados no futuro pelo cliente. Esse conjunto possui  $n$  blocos consecutivos, onde o primeiro bloco é determinado pelo modelo de comportamento de usuário apresentado em [De Vielmond et al. 2007]. A recuperação de blocos desse conjunto é feita segundo a política do bloco mais raro.

Na ocorrência de um salto do ponto corrente de visualização, a *janela de playback* pode ser atualizada ou não. Isso depende da nova posição para a qual se deu o salto. Seja  $d_{salto}$  a posição após o salto, e  $d_{inicial}$  e  $d_{final}$  a primeira e a última posição da *janela de playback*, respectivamente. Se  $d_{salto}$  não faz parte da *janela de playback* atual, ou seja,  $d_{salto} > d_{final}$  ou  $d_{salto} < d_{inicial}$ , a *janela de playback* é deslocada até que sua primeira posição (novo  $d_{inicial}$ ) venha a ser  $d_{salto}$ . Os próximos  $n$  blocos subsequentes constituirão a nova *janela de playback*. Caso contrário, se  $d_{inicial} \leq d_{salto} \leq d_{final}$ , a janela permanece na mesma posição, e diz-se que houve um salto interno, eliminando a necessidade de sua atualização.

A Figura 1(b) ilustra a operação do protocolo BIVoD. São apresentados quatro instantes de tempo diferentes. No instante  $t_0$ , o usuário acabou de chegar ao enxame e iniciou a recuperação para, assim que possível, começar a reproduzir o vídeo. A *janela de playback*, com tamanho igual a quatro blocos, contém inicialmente os blocos 1, 2, 3 e 4, sendo que o bloco 1 já foi pedido; enquanto que a *janela de previsão*, com tamanho igual a 3 blocos, contém os blocos 17, 18 e 19. O ponto corrente de *play* se encontra no bloco 1, o primeiro a ser reproduzido.

No instante de tempo  $t_1$ , o bloco 1, já recuperado, é reproduzido e o ponto de *play* é deslocado para o bloco seguinte. A *janela de playback* é deslocada para a direita porque o bloco 1 é o primeiro bloco da janela. Em seguida, o bloco 18 da *janela de previsão* é pedido, por ser o mais raro dentro dela. No instante  $t_{10}$ , o *play* está sobre o bloco 4, a ser recuperado e, em seguida, reproduzido. Novamente a *janela de previsão* é selecionada, sendo realizado o pedido do bloco 21. No instante  $t_{12}$ , o usuário executa um salto para frente, mais especificamente para o bloco 16. Portanto, o ponto de *play* é atualizado para esse ponto e a *janela de playback* também, fazendo agora parte dessa janela os blocos 16, 17, 18 e 19. Porém, observe que somente o bloco 16 ainda não foi recuperado e, assim que esse for recuperado, a janela se deslocará completamente para a direita, até o próximo bloco ausente. Ao mesmo tempo, a *janela de previsão* também é atualizada para os blocos 8, 9 e 10, sendo esses possivelmente os próximos blocos que o usuário deverá precisar.

Implementamos também a variante BIVoD-Buffer. Nessa variante, é apenas acres-

centado um *buffer* para o armazenamento de  $x < n$  blocos. A finalidade do *buffer* é de evitar que ocorra um excessivo número de interrupções durante a reprodução, causadas pela ausência de blocos consecutivos. A reprodução inicial só é habilitada quando o *buffer* está cheio. Na ocorrência de falta de bloco, a reprodução é paralisada e somente retomada após o *buffer* estar novamente cheio.

### 4.3. Diferenças entre BIVoD e Hoffmann-Rodrigues-Leão

As principais diferenças entre essas duas propostas são pontuadas a seguir.

Para a proposta BIVoD, tem-se: (i) existem dois conjuntos (*janela de previsão* e *janela de playback*) de onde os blocos são recuperados de forma alternada de cada um deles; (ii) quando da ocorrência de um salto, a *janela de playback* somente será deslocada se o ponto do salto for externo a ela; caso contrário, permanece na mesma posição; (iii) o cliente pode possuir um *buffer* (BIVoD-Buffer).

Por outro lado, para a proposta de Hoffmann-Rodrigues-Leão, tem-se: (i) existem três conjuntos (baixa prioridade, *janela de previsão* e *janela de playback*) de onde os blocos são recuperados com determinada probabilidade; (ii) quando ocorre um salto, a *janela de playback* é sempre deslocada para o início do salto; (iii) não é considerado *buffer* do lado do cliente.

## 5. Avaliação de performance

### 5.1. Métricas e Cargas

As métricas utilizadas nos experimentos estão definidas na Tabela 1. Quanto às cargas dos experimentos, são consideradas dois tipos: cargas reais e cargas sintéticas. As cargas reais são *logs* de comportamento de usuários do sistema multimídia RIO, utilizado no curso do CEDERJ. Um usuário do sistema RIO pode executar as seguintes ações: *play*, *stop*, *pause*, *salto para frente* e *salto para trás*. As cargas sintéticas são geradas usando o modelo de comportamento de usuário apresentado em [De Vielmond et al. 2007]. Para gerar essas cargas sintéticas, foram utilizados 391 *logs* reais com sessões de 20–30 minutos [Hoffmann Filho, Luiz J. 2009]. No modelo de simulação, o comportamento dos usuários é representado através das cargas reais, e as cargas sintéticas são usadas pelo algoritmo para prever as ações interativas futuras do cliente.

As cargas são classificadas em função do nível de interatividade  $I$ , estimado como o número médio de requisições por sessão. A classificação utilizada é: interatividade alta ( $15 < I < 40$ ); interatividade média ( $5 < I < 16$ ); interatividade baixa ( $0 < I < 6$ ) e interatividade mista ( $0 < I < 40$ ). Para garantir um significativo espectro de análise, as cargas são estatisticamente diferentes entre si. As estatísticas obtidas para as cargas estão na Tabela 2, onde  $N$  é o número de requisições,  $I$  é o número médio de requisições por sessão (nível de interatividade),  $L$  é o tamanho médio do segmento (número médio de blocos visualizados em sequência, em unidades de tempo),  $Std(L)$  é o desvio padrão de  $L$ , e  $Coeff(L)$  é o coeficiente de variação de  $L$ .

### 5.2. Experimentos

Os resultados do modelo de simulação são obtidos usando a ferramenta Tangram-II [de Souza e Silva et al. 2009]. Esses resultados são a média de 10 execuções

**Tabela 1. Métricas.**

Métricas	Cálculo das métricas
Número médio de interrupções ( $D$ )	$D = (\sum_{i=1}^U D_i)/U$ , onde $D_i$ é o número de interrupções que ocorreram no usuário $i$ e $U$ é o total de usuários do enxame. Essas interrupções decorrem da ausência de blocos.
Tempo médio de retorno ( $TR$ )	$TR = (\sum_{i=1}^U TR_i)/U$ , onde $TR_i$ é o tempo médio de retorno relativo ao usuário $i$ e $U$ é o total de usuários do enxame. É o tempo necessário para retomar à reprodução após uma interrupção.
Tempo médio para iniciar reprodução ( $TI$ )	$TI = (\sum_{i=1}^U TI_i)/U$ , onde $TI_i$ é o tempo relativo ao usuário $i$ e $U$ é o total de usuários do enxame.
Taxa de Download ( $TxD$ )	$TxD = (\sum_{i=1}^U TxD_i)/U$ , onde $TxD_i$ é a taxa média de <i>download</i> referente ao usuário $i$ e $U$ é o total de usuários do enxame.
Taxa de Upload ( $TxU$ )	$TxU = (\sum_{i=1}^U TxU_i)/U$ , onde $TxU_i$ é a taxa média de <i>upload</i> referente ao usuário $i$ e $U$ é o total de usuários do enxame.
Tempo médio de download ( $TD$ )	$TD = (\sum_{i=1}^U TD_i)/U$ , onde $TD_i$ é o tempo relativo ao usuário $i$ e $U$ é o total de usuários do enxame.

**Tabela 2. Estatísticas.**

Estatística	Nível de interatividade							
	Alta		Média		Baixa		Mista	
	Real	Sintética	Real	Sintética	Real	Sintética	Real	Sintética
$N$	1752	1582	1205	1287	388	454	3346	2541
$I$	24.01	21.68	9.80	10.46	1.99	2.33	8.56	6.50
$L$	26.03	26.80	61.40	61.70	260.65	260.70	75.54	106.47
$Std(L)$	29	33	68	65	260	263	77	107
$CoeF(L)$	1.114	1.23	1.11	1.05	1.00	1.00	1.02	1.00

e possuem um intervalo de confiança de 90%, variando entre 2% e 30% em torno do valor reportado. Salvo informado diferentemente, é considerado um único cenário para todas as simulações, tendo os seguintes parâmetros: tamanho do objeto igual a 1800 s, número de *seeds* igual a 1, banda disponível para cada um dos *leechers* e para o *seed* (banda de *download* e *upload*) igual a 100 kB/s, população com tamanho igual a 50 usuários (*leechers*) e a taxa de chegada (distribuição de Poisson) dos usuários ao enxame igual a  $\lambda = 4$  e 0.008 usuários/s. Todos os usuários deixam o enxame assim que terminam de assistir ao seu vídeo.

Os valores atribuídos aos parâmetros das propostas foram selecionados a partir de um estudo realizado em [Hoffmann Filho, Luiz J. 2009] e [Hoffmann Filho et al. 2009]. Nesse estudo foram obtidos os valores que possibilitavam o melhor desempenho para cada uma das propostas. Esses valores foram então aqui utilizados, constituindo uma comparação mais justa, uma vez que todas as propostas passam a ter seus respectivos desempenhos otimizados. Para as propostas BiToS, Zhou-Chiu-Lui e Hoffmann-Rodrigues-Leão (*Previsão Mais Rara* e *Previsão Sequencial*), foi selecionado  $p = 0.8$ . Para a proposta de Hoffmann-Rodrigues-Leão, tem-se  $q = 0.5$ . O tamanho do conjunto de alta prioridade é igual a 144 blocos para todas as propostas. Para a proposta de Hoffmann-Rodrigues-Leão, BIVoD e BIVoD-Buffer, a *janela de previsão* é igual ao tamanho médio do segmento ( $L$ ).

### 5.2.1. Análise competitiva

Neste experimento são avaliados o desempenho de todas as propostas descritas na Seção 3 quando são variados o nível de interatividade (alta, média e baixa) e a taxa de chegada ( $\lambda = 4$  e  $0.008$  usuários/segundo). Opta-se por apresentar algumas métricas para  $\lambda = 4$  e outras para  $\lambda = 0.008$ , dado que observou-se que a tendência é a mesma para ambas as taxas de chegada. Os resultados estão na Figura 2.

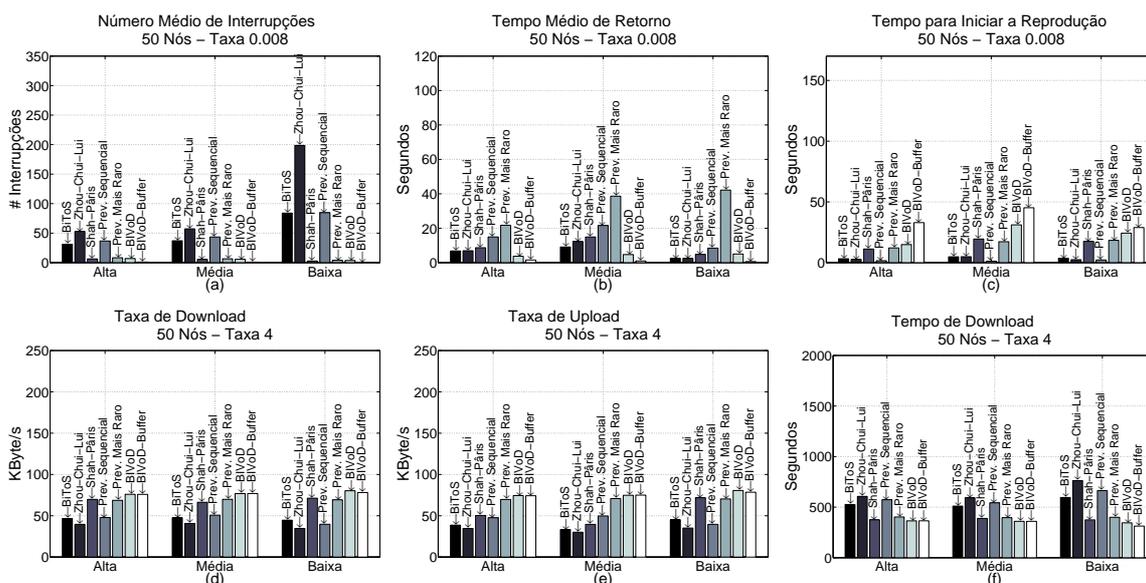


Figura 2. Comparação entre todas as propostas,  $\lambda = 4$  e  $\lambda = 0.008$ .

**Número médio de interrupções ( $D$ ):** A proposta de Zhou-Chiu-Lui apresenta o maior número de interrupções, seguida pelas propostas BiToS e *Previsão Sequencial*, independentemente do nível de interatividade ou da taxa de chegada dos usuários. As razões para essas propostas apresentarem desempenho inferior são: (i) utilização da política de seleção de blocos sequencial no conjunto de alta prioridade, que colabora para a ocorrência do problema de *toca e pára*, gerando várias interrupções consecutivas; (ii) recuperação de blocos do conjunto de baixa prioridade no lugar de blocos próximos de serem visualizados. As demais propostas, de Shah-Pâris, Previsão Mais Raro, BIVoD e BIVoD-Buffer, que utilizam a política de seleção do bloco mais raro, apresentam melhor desempenho. Por exemplo, na Carga Alta, a proposta BIVoD consegue uma redução de 86,91% em relação ao valor observado na proposta de Zhou-Chiu-Lui.

**Tempo médio de retorno ( $TR$ ):** Observa-se um bom desempenho das propostas BIVoD e BIVoD-Buffer. Esse resultado pode ser creditado à utilização da *janela de previsão*. A recuperação desses blocos auxilia na redução do tempo de espera, principalmente após um salto, onde pode ocorrer o deslocamento ou não da *janela de playback*, e como parte dos blocos dessa nova janela já foram recuperados pela *janela de previsão*, a tendência é que a recuperação dos blocos ausentes seja mais rápida. Por exemplo, na Carga Média, a proposta BIVoD-Buffer consegue uma redução de 95,46% em relação ao valor observado na proposta *Previsão Sequencial*.

**Tempo para iniciar a reprodução ( $TI$ ):** As propostas BIVoD, BIVoD-Buffer e de Shah-Pâris apresentam os maiores tempos. Por outro lado, as propostas BiToS e de Zhou-Chiu-Lui são as que iniciam mais rapidamente a reprodução do vídeo. Isso ocorre devido ao comentado a seguir. A proposta de Zhou-Chiu-Lui recupera os blocos do conjunto de alta prioridade sequencialmente, levando a um curto intervalo de tempo para iniciar. Já na proposta BiToS, no início da recuperação do vídeo, o número de cópias dos blocos deve ser igual, logo todos os blocos possuem a mesma *raridade*. Dada a mesma raridade, o protocolo dá maior prioridade aos blocos que estão mais próximos de serem tocados, o que forçará a recuperação dos primeiros blocos, caracterizando assim uma *pseudo* recuperação sequencial. As demais propostas, por utilizarem a política de seleção do bloco mais raro, inicialmente poderão ter um comportamento totalmente aleatório, o que poderá influenciar negativamente no desempenho dessa métrica. Por exemplo, na Carga Média, a proposta BIVoD-Buffer apresenta um valor mais de nove vezes maior que aquele obtido na proposta BiToS.

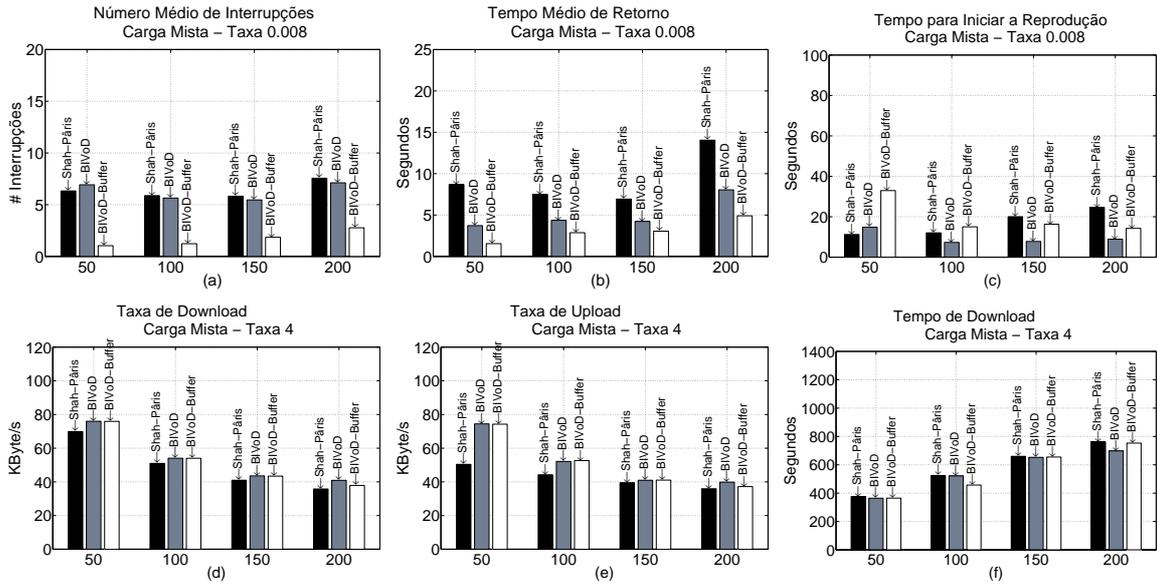
**Taxa de download ( $TxD$ ), Taxa de upload ( $TxU$ ) e Tempo de download ( $TD$ ):** As propostas BIVoD e BIVoD-Buffer alcançam as melhores taxas e, por conseguinte, apresentam a melhor utilização dos recursos do sistema. Isso deve-se à política de seleção de blocos mais raros. Por outro lado, as propostas de Zhou-Chiu-Lui e *Previsão Sequencial* alcançam as menores taxas devido à utilização da política de seleção sequencial de blocos. Por exemplo, na Carga Baixa e para a métrica  $TD$ , a proposta BIVoD-Buffer tem uma redução de 52,91% em relação ao valor obtido na proposta *Previsão Sequencial*.

### 5.2.2. Análise da escalabilidade

A partir dos resultados da seção anterior, foram escolhidos os três protocolos mais competitivos para realizar uma análise de como os mesmos se comportam com o aumento do número de usuários do enxame. Para isso, são realizados experimentos com populações de tamanho 50, 100, 150 e 200 usuários, todos com a carga mista, a qual julga-se prover um cenário mais próximo da realidade. Os resultados estão apresentados na Figura 3.

Considerando o *número médio de interrupções* ( $D$ ), a proposta BIVoD-Buffer claramente apresenta o melhor desempenho para todos os tamanhos de população. Por exemplo, para um população de 200 *peers*, a proposta BIVoD-Buffer consegue uma redução de 63,36% em relação ao valor observado para a proposta de Shah-Pâris. Nesse mesmo cenário, para o *tempo médio de retorno* ( $TR$ ), a proposta BIVoD consegue uma redução de 42,78% em relação ao valor observado para a proposta de Shah-Pâris. De forma interessante, apesar da proposta BIVoD-Buffer apresentar os melhores resultados também para a métrica  $TR$ , a diferença para a proposta BIVoD é geralmente menor, mostrando que, apesar da proposta BIVoD ter um *número médio de interrupções* ( $D$ ) maior, os usuários conseguem se recuperar rapidamente das interrupções.

Considerando agora o *tempo para iniciar a reprodução* ( $TI$ ), a proposta BIVoD é a que apresenta o menor valor na maioria dos cenários considerados. Por exemplo, para um população de 150 *peers*, a proposta BIVoD consegue uma redução de 61,16% em relação ao valor observado para a proposta de Shah-Pâris. Por fim, para as métricas *taxa de download* ( $TxD$ ), *taxa de upload* ( $TxU$ ) e *tempo de download* ( $TD$ ), as propostas BIVoD e BIVoD-Buffer apresentam, de forma geral, os valores mais otimizados. Por



**Figura 3. Avaliação comparativa entre as propostas de Shah-Pâris, BIVoD e BIVoD-Buffer com cenário carga mista e  $\lambda = 0.008$  e  $\lambda = 4$  usuários/segundo.**

exemplo, para uma população de 50 *peers* e para a taxa de upload ( $TxU$ ), a proposta BIVoD consegue um valor 47,70% superior àquele obtido na proposta de Shah-Pâris.

Em síntese, pode-se concluir que o desempenho das propostas BIVoD e BIVoD-Buffer é superior ao desempenho da proposta de Shah-Pâris. É importante mencionar que foram realizados experimentos em que a proposta de Shah-Pâris utiliza um *buffer*. Os resultados obtidos ao se comparar essa proposta e a proposta BIVoD-Buffer são bem semelhantes àqueles observados na comparação entre a proposta de Shah-Pâris e BIVoD, ambas sem *buffer*. Da mesma forma, análises sobre o impacto da escolha do tamanho das janelas no desempenho das propostas também foram realizadas. Detalhes sobre esses experimentos estão em [Hoffmann Filho, Luiz J. 2009].

### 5.2.3. Análise de equidade

Uma importante característica dos protocolos de distribuição de vídeo, que utilizam a arquitetura P2P, é o nível de equidade que o protocolo mantém entre os *peers* que participam da recuperação/distribuição de um vídeo. Para determinar o grau de equidade de cada proposta foram utilizadas as seguintes medidas: a média ( $\bar{x}$ ); a variância ( $\sigma^2$ ); e a diferença entre o valor máximo e mínimo ( $d$ ). Essas medidas foram calculadas para todas as métricas definidas na Seção 5.1. Foram escolhidas duas das métricas para apresentar os resultados obtidos: *tempo médio de retorno* ( $TR$ ) e *taxa de download* ( $TxD$ ). Esses resultados aparecem na Tabela 3. Ressalta-se que a tendência é a mesma para todas as métricas. A partir dos resultados apresentados, pode-se observar que as propostas BIVoD-Buffer e BIVoD promovem a melhor equidade entre os usuários e é devido ao uso do modelo de emulação de comportamento do usuário. Foi observado na proposta de Shah-Pâris que o valor de cada métrica obtida para cada usuário é influenciado pelo comportamento dele. Por exemplo, um usuário que realiza muitos saltos, certamente terá

um número maior de interrupções que um usuário que realiza poucos saltos. Logo, estes usuários apresentarão valores bem diferentes para as métricas pois estas sofrem forte influência do seu comportamento. No entanto, o uso do modelo de previsão, evita que a diferença do comportamento do usuário ocasione uma diferença na qualidade do vídeo.

**Tabela 3. Equidade.**

Carga	Protocolo	$TR(s)$			$TxD(kbyte/s)$		
		$\bar{x}$	$\sigma^2$	$d$	$\bar{x}$	$\sigma^2$	$d$
Alta	Shah-Pâris	8.68	244.73	77.35	69.86	171.21	46.31
	BIVoD	3.72	8.55	13.38	75.98	2.26	14.99
	BIVoD-Buffer	1.54	11.84	13.29	75.90	2.47	16.48
Média	Shah-Pâris	14.92	447.24	91.84	66.35	180.66	45.04
	BIVoD	4.69	14.53	17.04	76.91	0.30	1.98
	BIVoD-Buffer	0.98	2.81	7.89	76.95	0.40	2.32
Baixa	Shah-Pâris	4.77	28.50	22.68	71.29	92.92	45.51
	BIVoD	5.03	8.18	13.43	80.29	0.012	0.48
	BIVoD-Buffer	0.63	1.29	5.98	78.11	0.010	0.43

## 6. Conclusões e trabalhos futuros

Neste trabalho foi apresentado um novo protocolo, denominado *BitTorrent Interactive Video on Demand* (BIVoD), para acesso interativo de vídeo pré-armazenado. O desenvolvimento da proposta está fundamentado em dois conceitos. O primeiro é a utilização de uma variação do protocolo *BitTorrent* para VoD. O segundo é a utilização de um modelo de emulação de comportamento do usuário, com o objetivo de prever os blocos necessários no futuro pelo usuário devido a saltos que ele possa realizar. Além disso, também foi proposta uma variante desse novo protocolo, denominada BIVoD-Buffer, a qual diferenciou-se pelo uso de um *buffer* com o intuito de evitar interrupções quando da exibição do vídeo, bem como realizou-se uma extensa análise competitiva com diferentes métricas de performance e as mais recentes propostas da literatura.

Dentre as conclusões obtidas, destacam-se: (1) comparativamente a outras propostas da literatura, os protocolos BIVoD e BIVoD-Buffer levaram a otimizações de até mais de 90% para algumas das métricas de performance consideradas nos experimentos; (2) a política de seleção do bloco mais raro mostrou-se mais eficiente que a política de seleção sequencial de blocos, com relação à qualidade de exibição do objeto; (3) a seleção probabilística de blocos a partir de dois conjuntos, um com os blocos próximos a serem tocados e outro com todos os outros blocos, leva a um grande número de interrupções, pois blocos menos prioritários são recuperados em detrimento de blocos próximos de serem visualizados; (4) o uso de dois conjuntos de blocos só é vantajoso quando um deles contém os blocos prioritários e o outro contém blocos que serão solicitados em um futuro próximo. Daí, a importância do emprego de um modelo de comportamento de usuário.

Como trabalhos futuros, pretende-se considerar cenários com um maior número de nós e a heterogeneidade dos clientes e da rede. Além disso, visualiza-se a possível implementação de uma política de seleção de blocos adaptativa dentro de um mesmo conjunto de prioridade: gulosa para os primeiros blocos do conjunto e seleção do mais raro para os blocos restantes.

## Referências

- Carlsson, N. and Eager, D. L. (2007). Peer-assisted On-demand Streaming of Stored Media using BitTorrent-like Protocols. In *IFIP/TC6 Networking*, pages 570–581, Atlanta, GA, USA.
- Chu, Y., Rao, S. G., and Zhang, H. (2000). A Case for End System Multicast. In *ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, pages 1–12, Santa Clara, California, USA.
- Cohen, B. (2003). Incentives Build Robustness in BitTorrent. In *First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, CA, USA.
- de Souza e Silva, E., Figueiredo, D., and Leão, R. M. M. (2009). The TANGRAM-II Integrated Modeling Environment for Computer Systems and Networks. *ACM SIGMETRICS Performance Evaluation Review*, 36(4):45–65.
- de Souza e Silva, E., Leão, R. M. M., Santo, A. D., Azevedo, J. A., and Netto, B. C. M. (2006). Multimedia Supporting Tools for the CEDERJ - Distance Learning Initiative applied to the Computer Systems Course. In *22th ICDE World Conference on Distance Education*, pages 1–11, Rio de Janeiro, RJ, Brasil.
- De Vielmond, C. C. L. B., Leão, R. M. M., and de Souza e Silva, E. (2007). Um modelo HMM hierárquico para usuários interativos acessando um servidor multimídia. In *Simpósio Brasileiro de Redes de Computadores - SBRC*, volume I, Belém, Pará, Brasil.
- Hoffmann Filho, L. J., Rodrigues, C. K. S., and Leão, R. M. M. (2009). Acesso interativo para aplicações P2P de streaming de vídeo. In *XVII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 599–612, Recife, PE, Brasil.
- Hoffmann Filho, Luiz J. (2009). Algoritmos para Acesso Interativo em Aplicações de Vídeo P2P. Master's thesis, Universidade Federal do Rio de Janeiro - COPPE - Programa de Engenharia de Sistemas e Computação, Rio de Janeiro - RJ.
- Li, B., Xie, S., Keung, G., Liu, J., Stoica, I., Zhang, H., and Zhang, X. (2007). An Empirical Study of the Coolstreaming+ System. *IEEE Journal on Selected Areas in Communications*, 25(9):1627 – 1639.
- Netto, B. C. M., Azevedo, J. A., de Souza e Silva, E., and Leão, R. M. M. (2005). Servidor Multimídia RIO em Ensino à Distância. In *Proc. 6th International Free Software Forum*, Porto Alegre, RS, Brasil.
- Shah, P. and Pâris, J.-F. (2007). Peer-to-Peer Multimedia Streaming Using BitTorrent. In *IEEE International Performance, Computing, and Communications Conference - IPCCC*, pages 340–347, New Orleans, Louisiana, USA.
- Vlavianos, A., Iliofotou, M., and Faloutsos, M. (2006). BiToS: Enhancing BitTorrent for Supporting Streaming Applications. In *9th IEEE Global Internet Symposium*, Barcelona, Spain.
- Zhou, Y., Chiu, D. M., and Lui, J. C. S. (2007). A Simple Model for Analyzing P2P Streaming Protocols. In *IEEE International Conference on Network Protocols - ICNP*, pages 226–235, Beijing, China.