

# Protocolos de VoD baseados no BitTorrent\*

Carlo Kleber da Silva Rodrigues<sup>1</sup>  
Daniel Senna Panizzo<sup>2</sup>

## Resumo

O serviço de vídeo sob demanda (VoD) tornou-se bastante popular recentemente. Nesse contexto, este artigo explica o protocolo BitTorrent e os mais recentes protocolos de VoD que foram desenvolvidos tendo-o como base, além de análises competitivas entre esses protocolos bem como sugestões para trabalhos futuros que serão aqui mostradas.

**Palavras-chave:** Multimídia. Protocolos. Sistemas. Bittorrent. Streaming.

## 1 Introdução

Há algum tempo, com o advento da proliferação da Internet e o aumento de informações, iniciou-se uma busca por métodos eficientes de localização e replicação de conteúdo (RODRIGUES, 1999). As redes *peer-to-peer* (par-a-par), uma arquitetura de sistemas distribuídos caracterizada pela descentralização das funções na rede, surgiram como uma alternativa para este novo contexto (RODRIGUES, 2008).

---

\* Artigo recebido em: 17/12/2010

Artigo aprovado em: 05/ 09/2011

<sup>1</sup> Possui graduação em Engenharia Elétrica pela Universidade Federal da Paraíba (1993), mestrado em Sistemas e Computação pelo Instituto Militar de Engenharia (2000) e doutorado em Engenharia de Sistemas e Computação pela Universidade Federal do Rio de Janeiro (2006). Atualmente trabalha como engenheiro no Centro de Desenvolvimento de Sistemas (CDS) do Exército Brasileiro e é professor do Centro Universitário de Brasília – UniCEUB.

<sup>2</sup> Bacharel em Ciências da Computação, Tecnologia da Informação e Sistemas de Informação pelo UniCEUB; bacharel em Arquivologia pela UNB; atualmente profissionalmente como analista DW/BI.

Entre os principais benefícios das redes *peer-to-peer* (P2P), está a escalabilidade, que permite o uso dos recursos computacionais dos computadores pessoais conectados à rede. Com a evolução rápida e constante do desempenho dos computadores pessoais e popularização da banda larga, essas redes tornaram-se muito atrativas (DUARTE et al., 2008).

Apesar de nos últimos anos terem surgido diversas redes de compartilhamento de conteúdo *peer-to-peer* famosas como o Kazaa, eMule e Gnutella (LIAO et al., 2006), que focam na localização de conteúdo, a única aplicação de compartilhamento de conteúdo focada na replicação e amplamente utilizada é o BitTorrent.

O BitTorrent (COHEN, 2003) é um sistema de replicação de conteúdo pedaço por pedaço. Nesse sistema, quando vários pares estão realizando o *download* de um mesmo arquivo, eles fazem também o *upload* do arquivo um para o outro. É um método utilizado na busca da Eficiência de Pareto, que, na Ciência da Computação, significa um algoritmo de otimização local em que os pares das contrapartes verificam se podem melhorar suas partes juntos.

O desafio, hoje, reside em aprimorar o desempenho do protocolo BitTorrent, visto que ele já se tornou uma alternativa viável para a arquitetura cliente servidor, cuja capacidade de serviço não é escalável de acordo com o número de pares.

O restante deste texto tem a organização descrita a seguir. A Seção 1 traz os fundamentos necessários para o entendimento do protocolo BitTorrent. Além disso, esta seção também apresenta uma visão geral sobre os simuladores mais comuns que podem ser utilizados para a avaliação da performance de protocolos baseados no BitTorrent. A Seção 2 é voltada para a apresentação dos protocolos mais recentes desenvolvidos sob o paradigma do BitTorrent. Na Seção 3, encontram-se análises competitivas de performance dos protocolos discutidos na seção anterior. A Seção 4 encerra uma discussão crítica sobre o assunto em tela. Por fim, as conclusões do artigo e direções para trabalhos futuros.

## 2 Fundamentos

Nesta seção serão introduzidos os conceitos básicos para a compreensão do protocolo BitTorrent, seguidos por uma breve explanação sobre o seu funcionamento e apresentação de alguns simuladores de rede.

### 2.1 Conceitos

A terminologia utilizada pela comunidade *peer-to-peer*, em particular a da comunidade BitTorrent, ainda não é padronizada. Então, por motivo de melhor compreensão do artigo, definimos nesta seção os conceitos básicos utilizados no decorrer do artigo.

- **Peers:** São os membros de uma mesma rede de compartilhamento, na qual atuam tanto como clientes quanto como servidores.
- **Leechers e Seeds:** São os dois estados que podem ser atribuídos a um *peer*. Um *peer* é um *seed* quando possui o arquivo completo que está sendo compartilhado e apenas envia os pedaços desse arquivo para os *leechers*. Já um *leecher* é um *peer* que ainda não possui o arquivo por completo, ainda está recebendo os pedaços restantes do arquivo de outros *seeds* e *leechers* ao mesmo tempo em que envia para outros *leechers* os pedaços que já recebeu.
- **Tracker:** Para iniciar o *download* de um arquivo, primeiramente o *peer* deve obter o arquivo *torrent* para conectar-se com o *tracker* específico. Basicamente o *tracker* mantém uma lista de *leechers* e *seeds* que estão compartilhando o arquivo, são responsáveis por ajudar os *peers* a encontrarem uns aos outros.
- **Pedaços e subpedaços:** O BitTorrent corta os arquivos em pedaços de tamanho fixo para serem compartilhados, normalmente um quarto de *megabyte*. Esses mesmos pedaços são divididos em subpedaços. Normalmente de 16 *kilobytes*, para evitar a demora entre o envio dos pedaços. Quando um arquivo está sendo recebido por um *leecher*, um número de pedidos de pedaços é mantido em *pipeline*, normalmente 5.

- **Random first piece:** Política de seleção de pedaço utilizado pelo BitTorrent para iniciar a transferência de pedaços entre os *peers*. Como o *download* está no início e o *peer* ainda não possui nenhum pedaço para compartilhar, o primeiro pedaço é selecionado aleatoriamente; e, logo depois, a estratégia de compartilhamento muda para o algoritmo *rarest first*.
- **Rarest first algorithm:** Algoritmo importante de seleção de pedaços do BitTorrent que faz com que os pedaços do arquivo que são minoria entre os *peers* sejam os primeiros a serem replicados. Garante que os *peers* tenham sempre pedaços interessantes para compartilhar e que o envio de pedaços comuns seja adiado, além de minimizar o risco de que um pedaço em particular deixe de existir entre os *peers*.
- **Strict priority:** Outra política de seleção de pedaços faz com que sempre que um subpedaço é solicitado, os subpedaços restantes desse mesmo pedaço é solicitado antes de qualquer subpedaço de qualquer outro pedaço. Essa política garante que o envio completo dos pedaços seja o mais rápido possível.
- **End game mode:** Quando o *download* do arquivo se aproxima de seu fim, todos os subpedaços restantes são solicitados para todos os *peers*. À medida que os subpedaços são recebidos, os demais pedidos do mesmo subpedaço para outros *peers* são cancelados. Garante que o final do *download* seja concluído de forma rápida.
- **Choke algorithm:** Algoritmo utilizado na estratégia de seleção de *peers* para garantir níveis razoáveis de reciprocidade. O *choke algorithm* é diferente dependendo do estado em que se encontra o *peer*, se em *seed* ou *leecher*. No estado de *leecher*, pelo menos 4 *peers* interessados em receber pedaços do *leecher* devem ser desbloqueados. A decisão de quais *peers* devem ou não ser desbloqueados é baseada estritamente na taxa de *download* corrente. A cada 10 segundos o BitTorrent recalcula a taxa de *download* e os 3 *peers* com as melhores taxas são desbloqueados.

## 2.2 Protocolo BitTorrent

Como mencionado anteriormente, o BitTorrent é uma aplicação P2P que utiliza a banda dos *peers* que compartilham a rede para replicar o arquivo entre si. Seu funcionamento tem início quando um arquivo torrent é disponibilizado em um servidor e os usuários podem fazer o *download* desse arquivo *torrent* para participar da rede de compartilhamento do arquivo.

Ao fazer o *download* do arquivo *torrent* e juntar-se à rede, o *peer* recebe do *tracker* uma lista de outros *peers* que também estão compartilhando o arquivo. A partir daí, são utilizados os algoritmos de política de seleção de *peers* e pedaços do arquivo, como o *random first piece*, *rarest first*, *strict priority*, *choke alrithm* e *end game mode*. O *torrent* continua ativo na rede enquanto houver pelo menos uma cópia de cada pedaço do arquivo.

## 2.3 Simuladores de Rede

Durante a produção deste artigo, foram identificados alguns simuladores de rede que podem ser utilizados para avaliação dos protocolos baseados no BitTorrent. Esses simuladores são: Tangram (DE SOUZA E SILVA et al., 2009), Network Simulator, OctoSim (HWANG et al., 2008) e TorrentLab (MANSILHA et al., 2008).

O TorrentLab foi idealizado por pesquisadores brasileiros como um ambiente de simulação especificamente para o protocolo BitTorrent, tendo como intuito suprir uma demanda da comunidade científica por avaliações do BitTorrent. Porém, esse simulador ainda é apenas um conceito e ainda não foi produzido por seus idealizadores. O simulador OctoSim, desenvolvido pelos laboratórios de pesquisa da Microsoft, foi produzido para simular unicamente a rede BitTorrent. Nele é possível criar modelos com o comportamento dos *peers* (entradas, saídas, seleção de *peers* etc.) e com os algoritmos padrão do BitTorrent. No entanto, esse simulador não possui interface gráfica e carece de documentação a respeito de seu funcionamento, resumindo-se apenas a um arquivo “leia-me” dentro do arquivo de instalação e de comentários no decorrer do código do programa.

Já o Network Simulator e o Tangram são simuladores de redes, portanto são programáveis e podem simular diversos tipos de ambiente, não apenas o BitTorrent. Ambos possuem uma interface gráfica para que os pesquisadores visualizem de maneira melhor seus protótipos. O Tangram, como poderá ser observado mais adiante neste artigo, foi o simulador utilizado para realizar a análise competitiva de performance aqui apresentada.

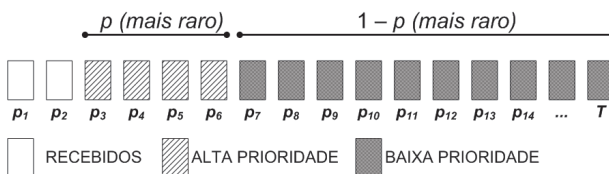
### 3 Protocolos

#### 3.1 Protocolo biToS

O protocolo *BitTorrent Streaming* (BiToS) (VLAVIANOS et al., 2006) é uma modificação do BitTorrent original para exibição sequencial de vídeo sem interatividade com o usuário. Baseia-se em alterações na política de seleção de pedaços, classificando-os em três conjuntos: recebidos, alta prioridade e baixa prioridade. Nesses conjuntos encontram-se, respectivamente, os pedaços que já foram recebidos, os que não foram recebidos e serão visualizados pelo usuário em breve e os que não foram recebidos e sua reprodução não está próxima.

Nesta modificação, cada *peer* solicita um pedaço do conjunto de alta prioridade com probabilidade  $p$  e um pedaço do conjunto de baixa prioridade com probabilidade  $(1-p)$ . A probabilidade  $p$  busca um equilíbrio entre a solicitação de pedaços que serão visualizados em breve e pedaços com visualização futura. A política de seleção de pedaços *rarest first* é utilizada independentemente do conjunto e os pedaços do conjunto de recebidos são aqueles que serão lidos e podem ser compartilhados com outros *peers*.

Figura 1 – Protocolo BiToS

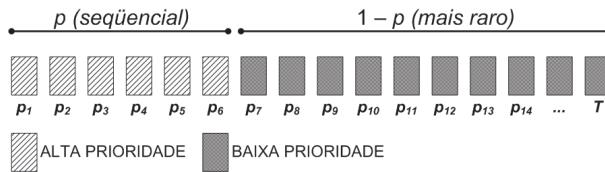


### 3.2 Protocolo de Zhou-Chiu-Lui

O protocolo de Zhou-Chiu-Lui (ZHOU et al., 2007), assim como o BiToS, baseia-se em alterações na política de seleção de pedaços, porém classifica-os em apenas dois: alta prioridade e baixa prioridade, que funcionam da mesma maneira que os conjuntos de mesmo nome definidos anteriormente no BiToS.

O uso da probabilidade  $p$  para solicitação de pedaços também se dá da mesma maneira que no protocolo BiToS, sendo  $p$  para o conjunto de alta prioridade e  $(1-p)$  para o de baixa prioridade. A diferença reside na utilização de uma política de seleção de pedaços sequencial no lugar do *rarest first* para os pedaços do conjunto de alta prioridade.

Figura 2 – Protocolo de Zhou-Chiu-Lui



Fonte: do Autor

### 3.3 Protocolo de Shah-Pâris

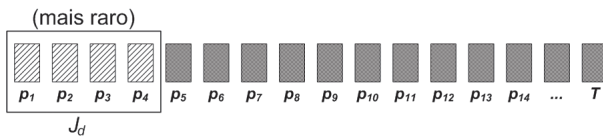
O protocolo de Shah-Pâris (SHAH; PÂRIS, 2007) implementa modificações na política de seleção de pedaços e na política de seleção de *peers* para distribuição de vídeo sob demanda sem interatividade.

Nesse protocolo, o vídeo é dividido em  $T$  pedaços e uma janela  $J_a$  é definida para compreender  $w$  pedaços em seqüência. Apenas os pedaços que estão compreendidos pela janela  $J_a$  podem ser solicitados em decorrência do mais raro. Para a exibição contínua do vídeo, existem duas condições possíveis para que esta janela deslize. A primeira condição é quando o primeiro pedaço da janela é recuperado, fazendo com que a janela deslize até que a primeira posição seja um pedaço que ainda não foi recuperado. A segunda condição estabelece que, após percorrido um

intervalo de tempo, denominado *playback delay*, a janela deve deslizar uma extensão de  $w$  pedaços.

A alteração na política de seleção *peers* estabelece que, a cada janela de  $w$  blocos recuperados,  $n$  *peers* da lista recebida do *tracker* são selecionados aleatoriamente e desbloqueados. Nos demais instantes de tempo, é utilizada a política de seleção de *peers*, original do protocolo BitTorrent.

**Figura 3** – Protocolo de Shah-Pâris



Fonte: do Autor

### 3.4 Protocolo de Hoffmann-Rodrigues-Leão

O protocolo de Hoffmann-Rodrigues-Leão (HOFFMANN FILHO; RODRIGUES; LEÃO, 2009) implementa alterações na política de seleção de pedaços para distribuição de vídeo sob demanda e se difere em relação aos protocolos anteriores no que se refere à interação do usuário, permitindo que sejam executadas ações como saltar para frente e para trás durante a execução do vídeo.

Nesse protocolo, os pedaços são classificados em dois conjuntos: janela de *playback* e  $B_p$ . A janela de *playback* compreende  $m$  pedaços em sequência, definidos como de alta prioridade. Já o conjunto  $B_p$  compreende  $T-m$  pedaços de baixa prioridade, sendo  $T$  o total de pedaços do vídeo. Dentro do conjunto  $B_p$  existe um subconjunto denominado janela de previsão, que possui  $k$  pedaços sequenciais, definidos de acordo com o modelo de comportamento de usuário apresentado em De Vielmond, Leão e De Souza e Silva (2007).

Além da classificação dos pedaços, são utilizadas probabilidades para selecionar de qual conjunto recuperar os pedaços. Com probabilidade  $p$ , é selecionada

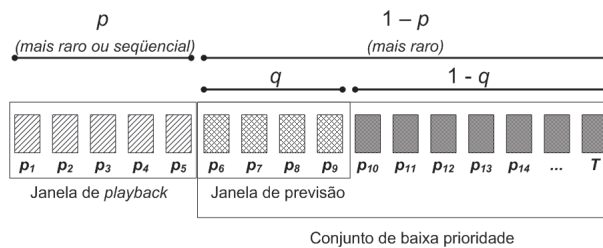


a janela de *playback*, com probabilidade  $(1-p)$  é selecionado o conjunto  $B_p$ . Se selecionado o conjunto  $B_p$ , com probabilidade  $q$ , são selecionados pedaços da janela de previsão e com probabilidade  $(1-q)$ , os pedaços fora dessa janela.

Quando selecionados pedaços para recuperação da janela de *playback*, são definidas variantes: Previsão Mais Raro, que utiliza a política de seleção de pedaços *rarest first*, e Previsão Sequencial, que utiliza uma política de seleção de pedaços sequencial. Já quando são selecionados pedaços da janela de previsão e do conjunto  $B_p$ , em ambos é utilizada a política de seleção de pedaços *rarest first*.

Quanto à interatividade, sempre que ocorre um salto na visualização do vídeo, a janela de *playback* é instantaneamente atualizada, fazendo com que a primeira posição dessa janela seja assumida pelo pedaço para o qual o usuário saltou. No caso de uma ação como essa, o conjunto  $B_p$  e a janela de previsão também são atualizados.

**Figura 4** – Protocolo de Hoffmann-Rodrigues-Leão



Fonte: do Autor

### 3.5 Protocolo BIVoD

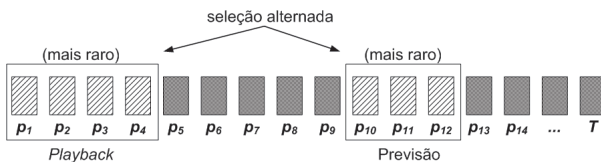
O protocolo BIVoD, assim como o Hoffmann-Rodrigues-Leão, faz alterações na política de seleção de pedaços e permite a interação do usuário enquanto o vídeo é executado. Nessa abordagem, os pedaços do vídeo são classificados em dois conjuntos: janela de *playback* e janela de previsão, nos quais a seleção dos pedaços a serem recuperados ocorre de forma alternada.

A janela de *playback* contém os pedaços de alta prioridade, ou seja, os pedaços que serão visualizados pelo usuário em breve. Essa janela possui  $m$  pedaços em sequência recuperados de acordo com a política de seleção de pedaços *rarest first*. Já a janela de previsão possui um conjunto de  $n$  pedaços consecutivos que serão visualizados pelo usuário em um futuro próximo, no qual o primeiro pedaço é determinado pelo modelo **de comportamento do usuário apresentado em De Vielmond, Leão e De Souza e Silva (2007)**. Essa última janela também utiliza a política de seleção de pedaços *rarest first*.

No caso da ocorrência de um salto no vídeo, a janela de *playback* pode ser atualizada ou não, dependendo do ponto em que foi feito o salto. Considerando  $d_{salto}$  a posição após o salto,  $d_{inicial}$  a primeira posição da janela de *playback* e  $d_{final}$  a última, se  $d_{inicial} \leq d_{salto} \leq d_{final}$ , a janela permanece da mesma posição e diz-se que ocorreu um salto interno, caso contrário, a janela é deslocada até que  $d_{inicial}$  seja igual a  $d_{salto}$ .

Nesse protocolo, também é implementada a variante BIVoD-Buffer. Esse *buffer* tem o intuito de evitar as pausas excessivas durante a execução do vídeo por falta de pedaços consecutivos. O *buffer* possui a capacidade de armazenar  $x < n$  pedaços e a reprodução inicial só é habilitada quando o *buffer* estiver cheio. Em caso de ausência de um pedaço, a execução do vídeo é paralisada e retomada somente quando o *buffer* estiver cheio novamente.

**Figura 5** – Protocolo BiVoD



Fonte: do Autor

#### 4 Análise competitiva de performance

Os resultados apresentados nesta seção constituem uma síntese obtida dos artigos estudados neste trabalho, especialmente aqueles contidos em Hoffmann Filho, Rodrigues e Leão (2009), em que é apresentada uma análise competitiva dos protocolos apresentados anteriormente. As métricas utilizadas para calcular o desempenho dos protocolos estão definidas no Quadro 1.

**Quadro 1** – Métricas

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

Fonte: do Autor

Os resultados apresentados foram obtidos utilizando a ferramenta de simulação Tangram-II. Esses resultados são a média de 10 execuções e possuem um intervalo de confiança de 90%. Foi considerado um único cenário para todas as simulações, tendo os seguintes parâmetros: tamanho do objeto igual a 1800s, nú-

mero de *seeds* igual a 1, banda disponível para cada um dos *leechers* e para o *seed* (banda de *download* e *upload*) igual a 100 *kbps*, 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. Sendo que todos os usuários deixam o enxame assim que terminam de assistir ao seu vídeo.

Em relação ao Número Médio de Interrupções (*D*), as propostas de Zhou-Chiu-Lui e BiToS apresentam o maior número de interrupções. Devido ao fato de utilizarem a política de seleção de pedaços sequencial no conjunto de alta prioridade e/ou de recuperarem pedaços do conjunto de baixa prioridade ao invés de pedaços que estão próximos de serem visualizados, o problema de *toca e para* torna-se frequente. Já as propostas de Shah-Pâris e BIVoD apresentam melhor desempenho, chegando, esta última, a uma redução de 86,91% em relação ao valor observado na proposta de Zhou-Chui-Lui.

Quanto ao Tempo Médio de Retorno (*TR*), devido à janela de previsão, a proposta BIVoD apresenta os melhores resultados. A recuperação de pedaços por essa janela auxilia na redução do tempo de espera para iniciar o vídeo após uma interrupção, principalmente depois de um salto. Porém, na métrica de Tempo para Iniciar a Reprodução (*TI*), os protocolos de Zhou-Chiu-Lui e BiToS apresentam melhor desempenho por utilizarem a recuperação sequencial de pedaços.

Encerrando a análise competitiva, a proposta BIVoD obtém os melhores resultados nas métricas Taxa de *Download* (*TxD*), Taxa de *Upload* (*TxU*) e Tempo de *Download* (*TD*), apresentando ótimo desempenho na utilização dos recursos do sistema. Desempenho alcançado, principalmente, devido à política de seleção dos pedaços mais raros. Por outro lado, a proposta de Zhou-Chiu-Lui alcança as menores taxas devido à política de seleção de pedaços sequencial.

Foi realizada também uma análise de escalabilidade para verificar como os protocolos se comportam com o aumento no número de usuários na rede de compartilhamento do arquivo. Buscando prover um cenário próximo do real, a análise foi feita com cargas mistas e com populações de tamanho 50, 100, 150 e 200 usuários. O resultado geral foi que a proposta BIVoD apresenta o melhor desempenho,

com melhorias de até, aproximadamente, 60% em algumas métricas, se comparada com a proposta de Shah-Pâris, que apresenta o segundo melhor resultado.

Por fim, para determinar o grau de equidade que os protocolos mantêm entre os *peers* que participam da recuperação/distribuição do vídeo, foi realizada uma análise de equidade. Utilizando as medidas de média, variância e diferença entre o valor máximo e mínimo de todas as métricas definidas na Tabela 1, mais uma vez a proposta BIVoD apresentou os melhores resultados. Essa melhor equidade entre os usuários pode ser atribuída ao uso do modelo de emulação de comportamento do usuário.

## 5 Discussão crítica

Os protocolos para transmissão de mídia contínua para VoD, usando arquitetura P2P, especificamente o protocolo BitTorrent, encontram-se em um estágio de desenvolvimento avançado.

A política de seleção de pedaços do algoritmo BitTorrent tem sido permanentemente alterada nas propostas mais recentes. Essa alteração no algoritmo visa receber rapidamente uma determinada quantidade de pedaços do início do vídeo para que a sua visualização seja a mais breve possível.

Por exemplo, na proposta BiToS e Zhou-Shiu-Lui, a divisão do arquivo foi feita em conjuntos para priorizar os pedaços do início do arquivo ao mesmo tempo em que aproveita a vantagem do algoritmo BitTorrent, que é receber pedaços de diferentes partes do arquivo. As propostas de Hoffmann-Rodrigues-Leão e BIVoD, ao invés de dividirem o arquivo inteiro em dois conjuntos, utilizaram duas janelas para limitar o conjunto de pedaços que serão solicitados, o que aumentou o desempenho das propostas e, ainda, sugeriu um ponto no qual os usuários provavelmente realizarão um salto no vídeo, acelerando o processo de retomada do *playback* em caso de interrupção.

Quanto à política de seleção de *peers*, houve pouca intervenção das propostas. Isso pode ser devido à dificuldade de implementar alterações no algoritmo ou mes-

mo do desempenho já satisfatório do algoritmo atual. Em especial, apenas a proposta de Shah-Pâris modificou o algoritmo, adicionando mais um instante, durante a recuperação dos pedaços, em que devem ser selecionados novos *peers* aleatoriamente.

Do exposto, pode-se afirmar que os estudos posteriores deverão concentrar-se na política de seleção de pedaços e, mais ainda, na política de seleção de *peers*. Em específico, é preciso buscar uma maior eficiência na seleção e recuperação dos pedaços mais próximos ao *playback*, bem como a realização de estudos mais aprofundados nas políticas de seleção de *peers* para verificar se o *choke algorithm* é realmente o suficiente quando se trata de VoD.

## 6 Conclusão

Neste trabalho, foram apresentados os principais conceitos para compreensão do funcionamento do protocolo BitTorrent e as atuais propostas que o modificam para transmissão de vídeo sob demanda (VoD). Foram apresentadas também análises competitivas de desempenho dessas propostas em um ambiente de simulação.

Dentre as propostas aqui analisadas, a do protocolo BIVoD apresenta o melhor desempenho. Os principais motivos foram: (1) esse protocolo utiliza uma política de seleção do pedaço mais raro, que se mostrou mais eficiente que a seleção sequencial de pedaços, então utilizada por outros protocolos; (2) esse protocolo não utiliza uma seleção probabilística de pedaços, empregada por outros protocolos, o que o torna menos complexo, redundando em menor *overhead*; (3) esse protocolo utiliza um modelo de comportamento do usuário, o qual permite prever quais pedaços serão visualizados em um futuro próximo.

Para trabalhos futuros, pretende-se realizar simulações com as propostas apresentadas, visando maior diversificação dos cenários de simulação. Nesse sentido, pode-se, por exemplo, verificar o desempenho das propostas quando do aumento do número de *peers* que ingressam na rede de compartilhamento do arquivo e o número de *seeds* iniciais. Isso permitirá, inclusive, detalhar melhor a análise da política de seleção de *peers*.

## Analysing BitTorrent-like protocols for VoD streaming

### Abstract

The video on demand (VoD) service has become very popular recently. Within this context, this article explains the BitTorrent protocol and the most recent VoD protocols that have been developed based on it. Additionally, competitive analyses between these recent protocols as well as suggestions for future work are here included.

**Keywords:** Multimedia. Protocols. Systems. Bittorrent. Streaming.

### Referências

COHEN, B. Incentives build robustness in BitTorrent. In: WORKSHOP ON ECONOMICS OF PEER-TO-PEER SYSTEMS, 1., 2003, Berkeley, USA. *Proceedings...* Berkeley, USA: Berkeley, 2003.

VIELMOND, C. C. L. B. de; LEÃO, R. M. M.; SILVA, E. de Souza e. Um modelo HMM hierárquico para usuários interativos acessando um servidor multimídia. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES – SBRC, 2007, Belém, PA. *Anais...* Belém, 2007.

DUARTE, O. C. M. B. et al. Distribuição de vídeo sobre redes par-a-par: arquiteturas, mecanismos e desafios, In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES – SBRC, 2008, Rio de Janeiro. *Minicursos. Anais...* Rio de Janeiro: Minicursos, 2008.

HOFFMANN FILHO, L. J. H.; RODRIGES, C. K. S.; LEÃO, R. M. M. Acesso interativo para aplicações P2P de streaming de vídeo. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS – SBRC, 27., 2009, Recife. *Anais...* Recife, 2009.

HOFFMANN FILHO, L. J.; RODRIGES, C. K. S.; LEÃO, R. M. M. BIVoD: Um novo protocolo para acesso interativo em sistema de vídeo sob demanda. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS – SBRC, 28., 2010, Gramado. *Anais...* Gramado, 2010.

HWANG, K. W.; MISRA, V.; RUBENSTEIN, D. *Stored media streaming in BitTorrent-like P2P networks*, 2008. Disponível em: <[http://dna-pubs.cs.columbia.edu/citation/paperfile/168/stored\\_media\\_streaming.pdf](http://dna-pubs.cs.columbia.edu/citation/paperfile/168/stored_media_streaming.pdf)> Acesso em: 10 dez. 2010.

LIAO, W.; PAPADOPOULOS, F.; PSOUNIS, K. Na efficient algorithm for resource sharing in peer-to-peer networks. *Lecture Notes in Computer Science*, Coimbra, v. 3979, p. 592-605, 2006.

MANSILHA, R. B.; BARCELLOS, M. P.; BRASILEIRO, F. V. TorrentLab: um ambiente para avaliação do protocolo BitTorrent. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS – SRBC, 26., 2008, Rio de Janeiro. *Anais...* Rio de Janeiro, 2008. p. 357-370.

RODRIGUES, C. K. S. *O problema do caminho mais rápido*. 1999. 200. Dissertação (Mestrado) – Ciências da Informação, Instituto Militar de Engenharia, Rio de Janeiro, 1999.

RODRIGUES, C. K. S. . Escalabilidade de Sistemas Interativos de Vídeo sob Demanda na Internet. *C & T: Revista Militar de Ciência e Tecnologia* , v. 25(1), p. 17-34, 2008.

SHAH, P.; PARIS, J.-F. Peer-to-peer multimedia streaming using bittorrent. IN *Ieee International Performance, Computing, and Communications Conference – IPCCC*. New Orleans, 2007, p. 340–347.

SILVA, E. de Souza e; LEÃO, R. M. M. The TANGRAM-II. Integrated modeling enviroment for computer systems and networks. *ACM SIG-METRICS: performance evaluation review*, New York, v. 36, n. 4, p. 45-65, 2009.

VLAVIANOS, A. ; ILIOFOTOU, M.; FALOUTSOS, M. BiToS: Enhancing BitTorrent for supporting streaming applications, In : IEEE GLOBAL INTERNET SYMPOSIUM, 9., 2006, Barcelona, Spain. *Proceedings...* Barcelona, 2006.

ZHOU, Y.; CHIU, D. LUI, J. C. S. A Simple model for analyzing P2P streaming protocols. In: IEEE INTERNATIONAL CONFERENCE ON NETWORK PROTOCOLS – INCP, 2007, Beijing, China. *Proceedings...* Beijing, 2007. p. 226-235.