

# Novos protocolos baseados no bittorrent para *streaming* de objetos multimídia na internet\*

## *Designing novel bittorrent-like protocols for more efficient multimedia streaming over internet*

Carlo Kleber da Silva Rodrigues<sup>1</sup>  
Ananda Gorck Streit<sup>2</sup>

### Resumo

Este artigo propõe dois novos protocolos para sistemas de streaming de objetos multimídia: BitTorrent Streaming Sequencial (BISS) e BitTorrent Streaming Interativo (BISI). Ambos são baseados no paradigma BitTorrent e um deles é especialmente desenvolvido para reprodução em ambientes interativos. Além disso, são revisados simuladores que podem ser utilizados para a avaliação da eficiência desses protocolos. Por meio de comparações e análises com outras propostas da literatura, o BISS e o BISI demonstram ser bastante competitivos, mantendo um compromisso satisfatório entre os requisitos de tempo e a diversidade de blocos do sistema.

**Palavras-chave:** BitTorrent. Streaming. Peer-to-Peer. Multimídia.

### Abstract

This article presents two novel protocols for multimedia streaming: BitTorrent Streaming Sequencial (BISS) and BitTorrent Streaming Interativo (BISI). They are both based on the BitTorrent paradigm and one of them is specially designed for interactive access to streaming systems. Furthermore, the main simulators that can be used to evaluate these protocols are presented and reviewed. By means of analysis as well as comparisons with other works in the literature, we may conclude that the novel protocols seem to be very competitive, since they maintain a good compromise between time constraints and the diversity of data blocks in the system.

**Keywords:** BitTorrent. Streaming. Peer-to-Peer. Multimedia.

---

\* Artigo recebido em 14/05/2012

Aprovado em 10/09/2012

<sup>1</sup> Graduado 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). Atua como engenheiro no Centro de Desenvolvimento de Sistemas (CDS) do Exército Brasileiro e é professor do Centro Universitário de Brasília – UniCEUB.

<sup>2</sup> Aluna de graduação do Curso de Bacharel em Ciência da Computação do UniCEUB e bolsista do CNPq em Projeto de Iniciação Científica.

## 1 Introdução

Não é possível garantir que aplicações baseadas em streaming na internet sejam imunes a problemas de atraso e de interrupções, ainda mais quando o conteúdo a ser distribuído é acessado por um grande número de usuários simultaneamente. Para aumentar a escalabilidade dessas aplicações, trabalhos recentes (CARLSSON; EAGER, 2007; HOFFMANN; RODRIGUES; LEÃO, 2011; VLAVIANOS; ILIOFOTOU; FALOUTSOS, 2006; ZHOU; CHIU; LUI, 2007) têm considerado a alternativa de adaptação do protocolo BitTorrent (COHEN, 2003) para suportar streaming.

O BitTorrent é um protocolo Peer-to-Peer (P2P) que incentiva a cooperação de usuários que estejam fazendo o download de um mesmo arquivo. Sua concepção é baseada em duas políticas, a saber: política de seleção de blocos de dados e política de seleção de peers. A primeira diz respeito à escolha dos dados a serem recuperados para visualização, e a segunda relaciona-se à decisão de quem na rede pode fornecer esses dados. As propostas da adaptação desse protocolo para streaming consistem essencialmente na modificação dessas políticas (BORGHOL et al., 2010; HUANG et al., 2008).

Pesquisas (MANSILHA; BARCELLOS; BRASILEIRO, 2008; SILVA; LEÃO, 2009) também têm se voltado para o desenvolvimento de simuladores. Isso porque as aplicações P2P oferecem grandes dificuldades para serem testadas em uma rede real devido à sua complexa arquitetura. Os simuladores surgem então como uma opção eficiente e justa para efeito de julgamento.

À luz do exposto, o presente trabalho tem duas contribuições importantes. A primeira é a proposição de dois novos protocolos baseados no BitTorrent para realizar streaming de objetos (arquivos) multimídia. Um desses protocolos, denominado BitTorrent Streaming Sequencial (BISS), é voltado para reprodução multimídia em ambientes não interativos. O outro, denominado BitTorrent Streaming Interativo (BISI), aplica-se a sistemas interativos. A segunda contribuição é a revisão dos principais simuladores que podem ser utilizados para avaliação de protocolos P2P, especialmente aqueles baseados no BitTorrent.

A organização do restante deste texto é descrita a seguir. A Seção 2 explica sucintamente o protocolo BitTorrent. Na Seção 3, são revistos os mais recentes traba-

lhos da literatura que focam na adaptação do protocolo BitTorrent para streaming de conteúdo multimídia. A finalidade desta revisão é tornar este texto o mais autocontido possível. Em seguida, a Seção 4 explica os dois novos protocolos: BISS e BISI. A Seção 5, por sua vez, discute os principais simuladores existentes na literatura. Por fim, conclusões e direcionamentos para trabalhos futuros.

## 2 Protocolo bittorrent

BitTorrent é um protocolo *Peer-to-Peer* (P2P) que busca replicar conteúdo de forma eficiente entre um conjunto de usuários (*peers*). Nesses sistemas, os *peers* organizam-se para receber (*download*) e transmitir (*upload*) um determinado arquivo, sendo responsáveis por manter o sistema ativo e por distribuir os custos de transmissão desse arquivo entre eles.

Um *peer* é denominado *leecher* quando ainda não possui o arquivo desejado por inteiro, podendo receber e transmitir conteúdo na rede, e é denominado *seed* quando já possui todo o arquivo, sendo responsável apenas pela transmissão de conteúdo.

No BitTorrent, o arquivo é dividido em blocos de mesmo tamanho que são requisitados pelo usuário em função da sua raridade. Assim, os blocos menos replicados do sistema são selecionados, diminuindo-se a probabilidade do arquivo estar incompleto na rede e aumentando-se a diversidade de blocos distribuídos entre os *peers*.

Para fazer o *download* de um determinado conteúdo, o usuário deve contatar o *tracker*. O *tracker* então registra o *peer* na lista de participantes do sistema de transmissão desse conteúdo e disponibiliza, de forma randômica, endereços IPs de outros *peers* presentes nesse sistema. A partir dessa lista de endereços, o usuário estabelece conexões e realiza requisições de blocos do arquivo.

Um usuário deve selecionar quais requisições de blocos recebidas serão atendidas. Essa seleção é realizada com base nas taxas de *download* oferecidas recentemente por cada *peer* requisitante. Assim, *peers* que ofereceram as maiores taxas de *download* são mais propensos para serem selecionados. Essa estratégia do BitTorrent é denominada *tit-for-tat* e é reconhecida por estimular a cooperação mútua (recíproca) entre os usuários e também por evitar *peers* egoístas, que não colaboram com a transmissão do arquivo pela rede.

### 3 Protocolos da literatura

As propostas apresentadas nesta seção são divididas em três categorias diferentes em função da estratégia de seleção de blocos utilizada, a saber: (i) Protocolos baseados em *janela única simples* (SAVOLAINEN; RAATIKAINEN; TARKOME, 2008; SHAH; PÂRIS, 2007); (ii) Protocolo baseado em *janela única ilimitada* (BORGHOL et al., 2010); e (iii) Protocolos baseados em *janela dupla simples* (HOFFMANN; RODRIGUES; LEÃO, 2011).

#### 3.1 Protocolos baseados em janela única simples

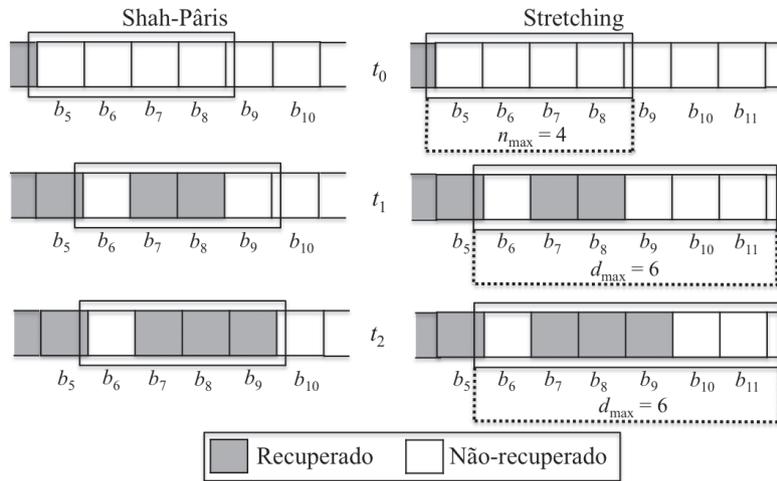
Os protocolos de Shah-Pâris (SHAH; PÂRIS, 2007) e Stretching (SAVOLAINEN; RAATIKAINEN; TARKOME, 2008) são voltados para a reprodução de um arquivo multimídia em um ambiente não interativo. Enquanto que no Stretching são realizadas modificações

apenas na política de seleção de blocos do BitTorrent, o protocolo de Shah-Pâris modifica tanto a política de seleção de blocos quanto a de seleção de *peers*.

Na política de seleção de blocos, uma janela deslizante  $J_d$  é introduzida. No protocolo de Shah-Pâris, essa janela é fixa e contém  $w$ , blocos em seqüência. No protocolo Stretching, a janela  $J_d$  não é fixa e possui dois limites máximos estabelecidos. O limite  $n_{max}$  define um número máximo de blocos não recuperados pelo usuário no interior da janela.

O limite  $d_{max}$  estabelece uma distância absoluta máxima entre o primeiro e o último bloco da janela. Tanto na proposta Stretching quanto na proposta de Shah-Pâris, os blocos da janela  $J_d$  são os únicos que podem ser requisitados pelo *peer* e são selecionados de acordo com a estratégia do mais raro.

Figura 1 - Política de seleção de blocos baseada em janela única simples



Fonte: do Autor

No instante  $t_0$  da Figura 1, a janela  $J_d$  compreende quatro blocos do arquivo. Com relação ao Stretching, o tamanho da janela nesse instante está relacionado com o limite  $n_{max}$ . No instante  $t_1$ , o usuário recupera três blocos e a janela desliza até o bloco  $b_6$  nos dois protocolos.

Isso ocorre porque o bloco  $b_5$ , o primeiro bloco da janela, é recuperado. A janela se mantém com um tamanho fixo no protocolo de Shah-Pâris. Por outro lado, no Stretching, a janela aumenta e compreende  $n_{max}$  mais dois blocos recuperados, atingindo o seu limite  $d_{max}$ . Finalmente, no tempo  $t_2$ , o usuário recupera mais um bloco.

Novamente a janela do protocolo de Shah-Pâris se mantém com o mesmo tamanho. No protocolo Stretching, a janela, que já havia atingido o limite  $d_{max}$ , também se mantém com o mesmo tamanho, porém passa a compreender três blocos recuperados e três blocos não recuperados.

Na política de seleção de *peers*, o protocolo Shah-Pâris determina que a cada janela de  $w$  blocos recuperados, um *peer* seleciona  $n$  *peers* randomicamente para desbloquear. Em todos os outros instantes de tempo, o *peer* segue exatamente a mesma política *tit-for-tat* apresentada no BitTorrent.

### 3.2 Protocolos baseados em janela única ilimitada

O protocolo Adaptive (BORGHOL et al., 2010) foi desenvolvido para suportar *streaming* de objetos multimídia em ambientes não interativos. Nesse protocolo, a estratégia *tit-for-tat* utilizada para selecionar os *peers* é mantida, alterando-se apenas a política de seleção de blocos do protocolo BitTorrent.

Os blocos do arquivo multimídia são selecionados no interior de uma janela deslizante  $w$  que adapta o seu tamanho de acordo com o desempenho alcançado pelo usuário no progresso de *download*. Com probabilidade  $p$ , essa seleção ocorre de acordo com a estratégia do mais raro. Porém, com probabilidade  $(1 - p)$ , os blocos são recuperados na ordem sequencial.

A cada seleção de blocos, a janela pode alterar o seu tamanho de acordo com a seguinte expressão:

$$w = \max [k (d - r - q), 0] + w_{\min}$$

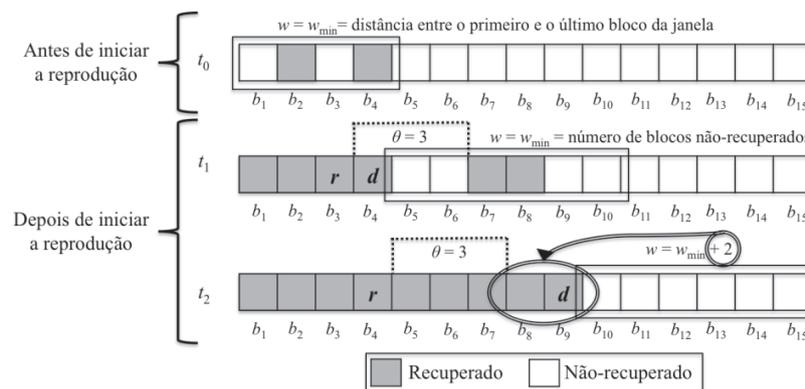
onde: a variável  $d$  corresponde ao último bloco contíguo

recuperado; a variável  $r$  é o ponto de reprodução do objeto;  $q$  é o limite mínimo de blocos contíguos estabelecidos;  $k$  é o coeficiente que determina o quanto a janela vai aumentar com relação à diferença  $(d - r - q)$ ; e  $w_{\min}$  representa o valor mínimo da janela.

Antes da reprodução do arquivo iniciar, a janela assume o seu tamanho mínimo, conforme exemplificado no instante  $t_0$  da Figura 2. Nesse momento, o tamanho mínimo da janela é determinado pela distância entre o primeiro e último bloco da janela. Porém, assim que a reprodução do arquivo inicia, o tamanho mínimo da janela passa a estar relacionado com o número de blocos não recuperados pelo usuário.

No instante  $t_1$ , por exemplo, a janela engloba quatro blocos não recuperados e, portanto, mesmo compreendendo seis blocos no seu interior, está no seu tamanho mínimo. No instante  $t_2$ ,  $(d - r)$  é maior que  $\theta$ . Assim, o tamanho da janela é igual a  $(d - r - q) = 2$ , somado com o valor mínimo  $w_{\min}$ .

Figura 2 - Política de seleção de blocos baseada em janela única ilimitada



Fonte: do Autor

### 3.3 Protocolos baseados em janela dupla simples

O BitTorrent Interactive Protocol - The First (BIP-F) e o BitTorrent Interactive Protocol - The Second (BIP-S) (HOFFMANN; RODRIGUES; LEÃO, 2011) são voltados para a visualização de arquivos multimídia em ambientes interativos. A única modificação realizada é na política de seleção de blocos do BitTorrent original.

No BIP-S, a seleção de blocos ocorre no interior de duas janelas deslizantes de tamanho fixo. Uma delas é a janela de reprodução, que engloba os blocos que acompanham o ponto de reprodução do arquivo. A outra é a ja-

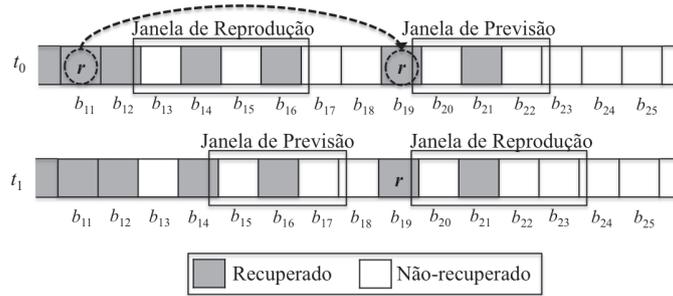
nela de previsão, que possui blocos que são determinados pelo modelo de comportamento do usuário apresentado por Vielmond, Leão e Silva (2007).

No BIP-F, por outro lado, a recuperação de blocos ocorre dentro e fora das janelas de reprodução e de previsão com diferentes probabilidades. Além disso, para a recuperação no interior da janela de reprodução, são estabelecidas duas variantes. A primeira utiliza a estratégia do mais raro, enquanto a segunda recupera os blocos de forma sequencial. Para o restante do arquivo, os blocos são selecionados em função da sua raridade.

A Figura 3 ilustra o comportamento das janelas de reprodução e de previsão quando uma ação interativa é executada. No instante  $t_0$ , o usuário realiza uma ação interativa e o ponto de reprodução  $r$ , que estava localizado

no bloco  $b_{11}$ , é atualizado para o bloco  $b_{19}$ . Dessa forma, no instante  $t_1$ , a janela de reprodução salta até esse ponto e a janela de previsão é atualizada de acordo com o modelo de comportamento do usuário.

Figura 3 - Política de seleção de blocos baseada em janela dupla simples



Fonte: do Autor

Na atualização de janelas, o protocolo BIP-S ainda apresenta mais uma particularidade. Nesse protocolo, a janela de reprodução apenas é atualizada quando uma ação interativa é executada e o novo ponto de reprodução não representa um bloco que já estava em seu interior.

## 4 Duas novas propostas

Nesta seção, são propostos dois novos protocolos baseados no BitTorrent para *streaming* de objetos multimídia. Cada um desses protocolos é apresentado por meio de uma descrição detalhada sobre o seu funcionamento e é comparado competitivamente com outros propostos na literatura.

### 4.1 Protocolo bittorrent streaming sequencial

#### 4.1.1 Operação

O protocolo BitTorrent Streaming Sequencial (BISS), como informado na Introdução, é voltado para *streaming* de objetos multimídia em ambientes não interativos. Nesse protocolo, são realizadas modificações apenas na política de seleção de blocos do BitTorrent original.

O arquivo é dividido em três conjuntos diferentes:  $A_p$ ,  $B_p$  e  $C_r$ . O conjunto  $A_p$  engloba os blocos do arquivo que ainda não foram recuperados pelo usuário e que serão reproduzidos em breve. Esse conjunto possui um tamanho  $n$  variável. Os blocos do conjunto  $C_r$  são aqueles que já foram recebidos pelo usuário. E os blocos do

conjunto  $B_p$  representam aqueles que ainda não foram recuperados pelo usuário e que não serão reproduzidos em breve.

A seleção de blocos é realizada apenas no conjunto  $A_p$  utilizando-se a estratégia do mais raro. Também é definida uma janela deslizante  $J_d$  que engloba todo o conjunto  $A_p$  e pode possuir blocos do conjunto  $C_r$ . Essa janela possui  $w$  blocos em seu interior, sendo  $w = n + q$ , onde  $n$  representa os blocos do conjunto  $A_p$ , e  $q$  representa os blocos recuperados que estão no interior da janela, ou seja, representa parte do conjunto  $C_r$ .

A janela  $J_d$  possui um valor máximo  $w_{max}$  e um valor mínimo  $w_{min}$ . O valor mínimo está diretamente relacionado ao tamanho do conjunto  $A_p$ , já que a janela não pode englobar menos blocos que  $n$ . Esses valores são adaptados a cada seleção de blocos, de acordo com as seguintes duas equações:

$$w_{max} = \max [k_{max} (d - r - q), 0] + v_m \text{ e } w_{min} = \max [k_{min} (d - r - q), 0] + v_m$$

onde: a variável  $d$  corresponde ao último bloco contíguo recuperado; a variável  $r$  é o ponto de reprodução do objeto;  $q$  é o limite mínimo de blocos contíguos estabelecidos;  $k_{max}$  e  $k_{min}$  são coeficientes que determinam o quanto os valores  $w_{max}$  e  $w_{min}$ , respectivamente, vão aumentar com relação a diferença  $(d - r - q)$ ; e  $v_m$  representa o valor mínimo das variáveis  $w_{min}$  e  $w_{max}$ .

Os limites  $w_{min}$  e  $w_{max}$  são adaptados em função do desempenho de *download* alcançado pelo usuário e

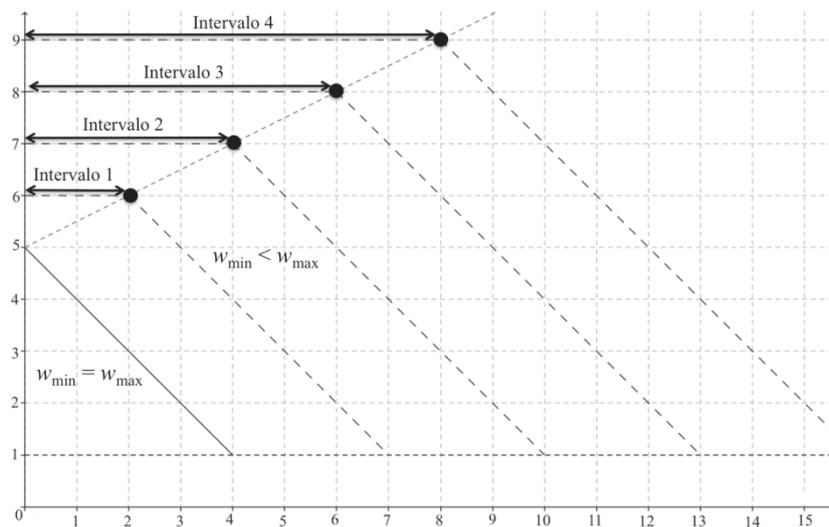
possuem valores diferentes de acordo com os coeficientes  $k_{\min}$  e  $k_{\max}$ . O coeficiente  $k_{\max}$  deve possuir um valor maior que o coeficiente  $k_{\min}$ . Isso porque  $w_{\min}$  não pode ultrapassar  $w_{\max}$ .

Definindo um valor mínimo  $v_m$  para os limites  $w_{\min}$  e  $w_{\max}$ , o usuário que não passar de  $q$  blocos consecutivos, a partir do ponto de reprodução, possuirá uma janela de tamanho  $w_{\min} = w_{\max} = v_m$ . Nessa situação, quando um bloco é recuperado em seu interior, a janela  $J_d$  irá se manter com o mesmo tamanho, e a quantidade de blocos do conjunto  $A_p$  diminuirá. Dessa forma, restringe-se a re-

cuperação de blocos que estão mais próximos do ponto de reprodução.

Por sua vez, no momento em que o usuário possui um bom desempenho no *download* do arquivo, os limites  $w_{\min}$  e  $w_{\max}$  recebem valores diferentes de acordo com os coeficientes  $k_{\min}$  e  $k_{\max}$ . Na verdade, o objetivo é adaptar o intervalo em que a janela pode variar de tamanho com relação ao desempenho do usuário. Dessa forma, estimula-se a recuperação de blocos diferentes e que não estão necessariamente mais próximos do ponto de reprodução do arquivo.

Figura 4 - Tamanho da janela  $J_d$  do BISS de acordo com os valores  $n$  e  $q$



Fonte: do Autor

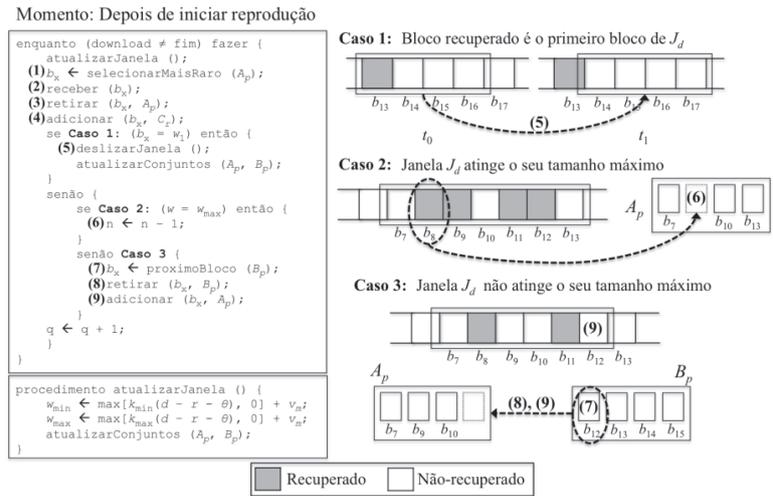
Na Figura 4, é apresentado um gráfico a fim de exemplificar o intervalo de variação do tamanho da janela  $J_d$  com relação aos limites  $w_{\min}$  e  $w_{\max}$ . Nesse exemplo,  $v_m = 5$  e os coeficientes  $k_{\max}$  e  $k_{\min}$  são, respectivamente, iguais a 3 e 1. Quando  $w_{\min} = w_{\max}$ , a janela  $J_d$  não pode variar de tamanho.

Quando  $w_{\min} < w_{\max}$ , a janela  $J_d$  passa a variar de tamanho em intervalos diferentes. No intervalo 1, varia entre seis e oito blocos. No intervalo 2, entre sete e onze blocos, e assim, sucessivamente, conforme demonstrado na figura. Observe que os pontos do gráfico representam o limite  $w_{\max}$  e as retas decrescentes representam a janela  $J_d$  que já atingiu  $w_{\max}$  e que, portanto, diminui o valor de  $n$  e aumenta o valor de  $q$  assim que são recuperados blocos no seu interior.

Antes da reprodução do objeto multimídia iniciar, os primeiros  $w$  blocos devem ser recebidos pelo usuário. Nesse momento, os valores de  $w_{\min}$  e  $w_{\max}$  são iguais. Dessa forma, prioriza-se a recuperação dos blocos iniciais e diminui-se o atraso de início de reprodução.

Assim que a reprodução inicia, o usuário continua fazendo o *download* do arquivo. A Figura 5 ilustra esse momento. Primeiro, é necessário atualizar os valores  $w_{\min}$  e  $w_{\max}$  da janela. Depois disso, o bloco mais raro do conjunto  $A_p$  é recebido pelo usuário no passo (1) do algoritmo. No passo (2), o usuário faz a requisição desse bloco e aguarda o seu recebimento. No passo (3), retira-se o bloco do conjunto  $A_p$  e no passo (4), adiciona-se esse bloco ao conjunto  $C_r$ . Em seguida, três casos diferentes podem ocorrer.

Figura 5 - Exemplos e algoritmos para o momento em que a reprodução do arquivo já foi iniciada no BISS



Fonte: do Autor

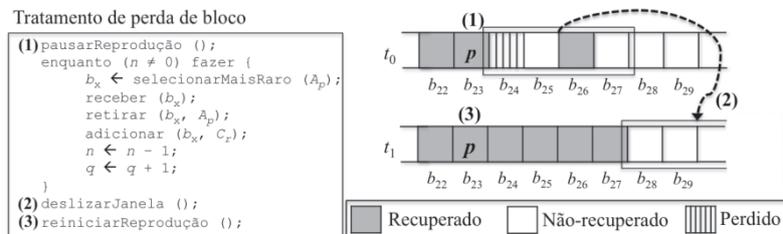
O Caso 1 ocorre quando o bloco que foi recuperado pelo usuário representa  $w_1$ , o primeiro bloco da janela  $J_d$ . No exemplo, o usuário recupera o bloco  $b_{13}$  no tempo  $t_0$  e, no tempo  $t_1$ , a janela desliza até que o seu primeiro bloco seja um que ainda não tenha sido recuperado pelo usuário, isto é, o bloco  $b_{14}$ . Após ocorrer o deslizamento, os conjuntos  $A_p$  e  $B_p$  são atualizados.

O Caso 2 ocorre quando a janela atinge  $w_{max}$ . No exemplo, o usuário recupera o bloco  $b_8$  e o conjunto  $A_p$  não recebe o próximo bloco do conjunto  $B_p$ , diminuindo de tamanho. Isso mantém o tamanho da janela  $J_d$  constante. Por fim, o Caso 3 ocorre quando a janela  $J_d$  aumenta o seu tamanho após recuperar um bloco superior a  $w_1$ . No passo (7) do algoritmo, o blo-

co  $b_{12}$ , o próximo bloco do conjunto  $B_p$ , é selecionado. Os passos (8) e (9) demonstram a transferência desse bloco para o conjunto  $A_p$ . Esse conjunto mantém o seu valor e a janela  $J_d$  aumenta de tamanho por conta do aumento de  $q$ .

Quando um bloco não é recuperado a tempo para ser reproduzido, o arquivo multimídia deve ser pausado. Ressalta-se que, nesse momento,  $w_{min} = w_{max}$ . De acordo com a Figura 6, o arquivo somente será reiniciado quando todos os blocos que ainda não tinham sido recuperados no interior da janela  $J_d$  são recebidos pelo usuário. Esse tratamento é importante porque previne que usuários com baixo desempenho no *download* tenham uma quantidade excessiva de interrupções.

Figura 6 - Exemplo e algoritmo para o tratamento de perda de bloco do BISS



Fonte: do Autor

#### 4.1.2 Tabela comparativa

A Tabela 1 compara as características dos principais protocolos baseados no BitTorrent e voltados para *streaming* de objetos multimídia em ambientes não inte-

rativos. Observa-se que apenas o protocolo de Shah-Pâris realizou alterações na política de seleção de *peers* do BitTorrent. Isso ocorre porque a política do BitTorrent ainda é uma estratégia bastante competitiva (CARLSSON; EA-

GER, 2007; LEGOUT; URVOY-KELLER; MICHIARDI, 2006; SHAH; PÂRIS, 2007).

Por outro lado, a política de seleção de blocos é alterada por todos os protocolos apresentados. Os protocolos de Shah-Pâris e Stretching realizam essa alteração por meio de uma janela deslizante simples que caracteriza o subconjunto do arquivo de onde os blocos são selecionados. Shah-Pâris é o único que possui uma janela fixa que não altera o seu tamanho. Já o protocolo Stretching

possui uma janela dinâmica com um tamanho máximo definido.

Os protocolos Adaptive e BISS, por sua vez, possuem uma janela adaptativa com um tamanho que é computado dinamicamente de acordo com o progresso de *download* do usuário. No protocolo Adaptive, a adaptação ocorre apenas no limite de tamanho mínimo da janela. Já o protocolo BISS é mais completo e adapta tanto o limite mínimo quanto o máximo da janela.

**Tabela 1** - Tabela Comparativa com os principais protocolos baseados no BitTorrent para *streaming* de objetos multimídia em ambientes não-interativos

Protocolo	Política de Seleção de Peers	Política de Seleção de Blocos	Tamanho da Janela	Limites de Tamanho Adaptados
Shah-Pâris	Alterada	Janela Única Simples	Fixo	-
Stretching	BitTorrent original	Janela Única Simples	Dinâmico e Limitado	-
Adaptive	BitTorrent original	Janela Única Ilimitada	Dinâmico e Não Limitado	Limite mínimo
BISS	BitTorrent original	Janela Única Limitada	Dinâmico e Limitado	Limite mínimo e Limite máximo

Fonte: do Autor

A limitação do tamanho da janela estimula a recuperação de blocos mais próximos do ponto de reprodução sem comprometer a diversidade de blocos no sistema. É possível perceber que o protocolo BISS apresentado neste trabalho agrega algumas características da política de seleção de blocos dos protocolos Stretching e Adaptive.

Porém, ao contrário do Stretching, o protocolo BISS não estabelece um limite máximo constante durante o *download* do arquivo e, assim, busca aumentar a diversidade de blocos apenas quando isso é vantajoso para o usuário. Além disso, também não permite que a janela aumente de tamanho indefinidamente como o Adaptive, diminuindo a probabilidade de ocorrerem interrupções durante a reprodução do arquivo multimídia. Isso torna o protocolo BISS potencialmente mais eficiente.

## 4.2 Bittorrent streaming interativo

### 4.2.1 Operação

O protocolo BitTorrent Streaming Interativo (BISI), conforme mencionado na introdução, é voltado para *streaming* de objetos multimídia em ambientes interativos. Nesse protocolo, foram realizadas modificações apenas na

política de seleção de blocos do BitTorrent original.

Os blocos do arquivo são classificados em três conjuntos:  $A_p$ ,  $C_r$  e  $C_p$ . Os blocos do conjunto  $A_p$  são aqueles que devem ser assistidos em breve pelo usuário. O conjunto  $C_r$  possui os blocos que já foram recebidos pelo usuário. Por fim, o conjunto  $C_p$  possui os blocos que estão previstos para serem selecionados pelo usuário no futuro, de acordo com o modelo de comportamento de usuário de De Vielmond et al. (2007). Tanto no conjunto  $A_p$  quanto no conjunto  $C_p$ , os blocos são selecionados de forma alternada de acordo com a estratégia do mais raro.

Além disso, duas janelas deslizantes são definidas: a janela de reprodução  $J_r$  e a janela de previsão  $J_p$ . A janela de reprodução engloba todo o conjunto  $A_p$  e pode possuir blocos do conjunto  $C_r$ . Essa janela compreende  $w_r = n_r + q_r$  blocos em seu interior, sendo  $n_r$  o número de blocos do conjunto  $A_p$ , e  $q_r$  o número de blocos que se encontram nessa janela e que já foram recuperados pelo usuário, isto é, parte do conjunto  $C_r$ .

A janela de previsão engloba todo o conjunto  $C_p$  e também pode possuir blocos do conjunto  $C_r$ . Essa janela

possui um tamanho  $w_p = n_p + q_p$ , onde  $n_p$  é o número de blocos do conjunto  $C_p$  e  $q_p$  é o número de blocos que se encontram nessa janela e que já foram recuperados, isto é, parte do conjunto  $C_r$ .

Essas duas janelas podem possuir tamanhos diferentes entre si. Considerando que o objetivo da janela de previsão é garantir que não ocorram interrupções quando o usuário executa uma ação interativa, essa janela deve recuperar a maior quantidade de blocos consecutivos antes que a ação seja executada. Dessa forma,  $J_p$  possui uma quantidade fixa de  $w_p$  blocos.

Por outro lado, a janela de reprodução tem como objetivo principal acompanhar a reprodução do objeto, recuperando os blocos subsequentes do ponto de reprodução antes que os seus prazos para *download* acabem. Assim, quando o usuário está alcançando um bom desempenho no *download* do arquivo, essa janela pode adaptar o seu tamanho a fim de proporcionar uma maior

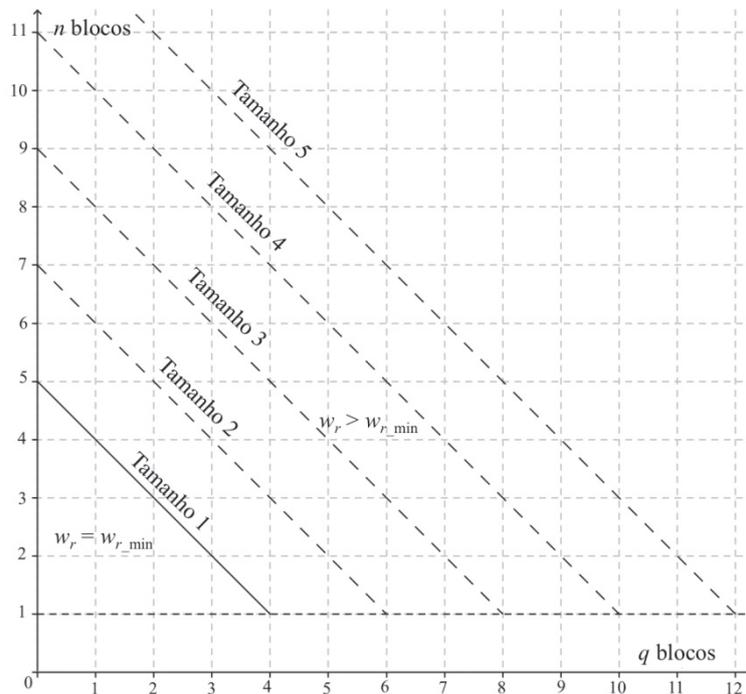
diversidade de blocos ao sistema. A adaptação da janela  $J_r$  ocorre segundo a expressão:

$$w_r = \max [k (d - r - q), 0] + w_{r\_min}$$

onde: a variável  $d$  corresponde ao último bloco contíguo recuperado; a variável  $r$  é o ponto de reprodução do objeto;  $q$  é o limite mínimo de blocos contíguos estabelecidos;  $k$  é o coeficiente que determina o quanto a janela  $J_r$  vai aumentar com relação a diferença  $(d - r - q)$ ; e  $w_{r\_min}$  representa o valor mínimo de  $J_r$ .

A Figura 7 ilustra o comportamento que a janela  $J_r$  pode apresentar durante o *download* do arquivo. Nesse exemplo,  $w_{r\_min} = 5$  e  $k = 2$ . A janela  $J_r$  pode ter tamanho igual a 1, quando  $w_r = w_{r\_min}$ , possuindo cinco blocos em seu interior. Pode ter tamanho igual a 2, quando  $w_r > w_{r\_min}$ , possuindo sete blocos em seu interior, ou ainda pode englobar nove blocos (tamanho da janela igual a 3). As retas decrescentes do gráfico mostram que  $n$  diminui e  $q$  aumenta de forma compensatória assim que são recuperados blocos no interior da janela  $J_r$ .

Figura 7 - Tamanho da janela  $J_r$  do BISI de acordo com os valores  $n$  e  $q$



Fonte: do Autor

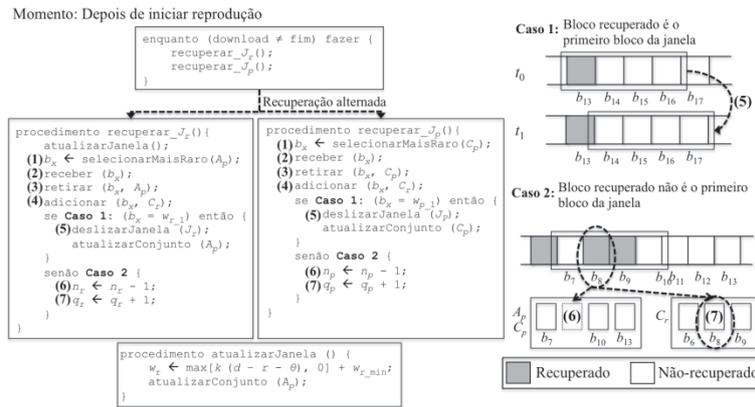
Antes da reprodução do objeto iniciar, os primeiros  $w_r$  blocos devem ser recebidos pelo usuário. Nesse momento,  $w_r = w_{r\_min}$ . Dessa forma, prioriza-se a recuperação dos blocos iniciais e diminui-se o atraso de início de reprodução. Quando a reprodução do arquivo inicia, a sele-

ção de blocos é realizada nos conjuntos  $A_p$  e  $C_p$  de forma alternada. A Figura 8 ilustra esse momento. Quando, por exemplo, a recuperação ocorre dentro de  $J_r$ , deve-se, em primeiro lugar, atualizar essa janela. Isso porque  $J_r$  possui um tamanho adaptativo  $w_r$ .

Em segundo lugar, o conjunto  $A_p$  é atualizado em função da alteração do tamanho da janela  $J_p$ , e o bloco mais raro desse conjunto é recebido pelo usuário no passo (1) do algoritmo. No passo (2), o usuário faz a requi-

sição desse bloco e aguarda o seu recebimento. No passo (3), retira-se o bloco do conjunto  $A_p$  e, no passo (4), adiciona-se esse bloco ao conjunto  $C_r$ . Em seguida, dois casos diferentes podem ocorrer.

Figura 8 - Exemplos e algoritmos para o momento em que a reprodução do arquivo já foi iniciada no BISI



Fonte: do Autor

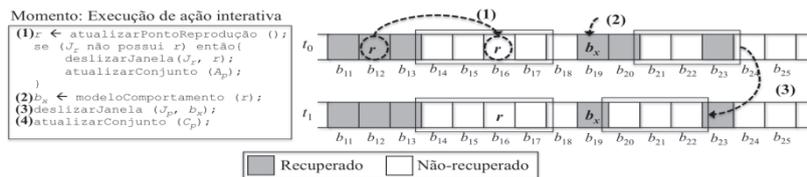
O Caso 1 ocorre quando o bloco recuperado representa  $w_{r-1}$ , ou seja, o primeiro bloco da janela  $J_r$ . Nessa situação, o passo (5) determina que a janela deve deslizar até que o seu primeiro bloco seja um que ainda não tenha sido recuperado pelo usuário. No exemplo, o usuário recupera o bloco  $b_{13}$  no tempo  $t_0$  e no tempo  $t_1$  a janela  $J_r$  desliza até o bloco  $b_{14}$ . Após ocorrer o deslizamento, o conjunto  $A_p$  é atualizado.

O Caso 2 acontece quando o bloco recuperado não é o primeiro da janela  $J_r$ . Nessa situação, aumenta-se  $q_r$  e diminui-se  $n_r$ . Feito isso, a recuperação passa a ser realizada na janela  $J_p$ . A única diferença no processo de recuperação dessa janela e da janela  $J_r$  é que não é necessário atualizar o seu tamanho, já que ele é fixo durante o *download* do arquivo inteiro.

Além disso, essa proposta prevê ações interativas durante a reprodução do arquivo. Assim, o usuário pode modificar a localização do ponto corrente de reprodução e passar a visualizar outro bloco do arquivo. De acordo com o passo (1) da Figura 9, o ponto de reprodução  $r$ , que estava localizado no bloco  $b_{12}$ , é atualizado para o bloco  $b_{16}$  no tempo  $t_0$ .

A partir desse salto, a janela de reprodução pode ou não ser atualizada e passar a englobar outros blocos. Isso depende da nova localização para a qual se deu o salto. No exemplo, o novo ponto de reprodução passou a ser um bloco que já era englobado pela janela  $J_r$ . Assim, no tempo  $t_1$ , a janela não atualiza a sua localização.

Figura 9 - Exemplo e algoritmo para o momento em que o usuário executa uma ação interativa no BISI



Fonte: do Autor

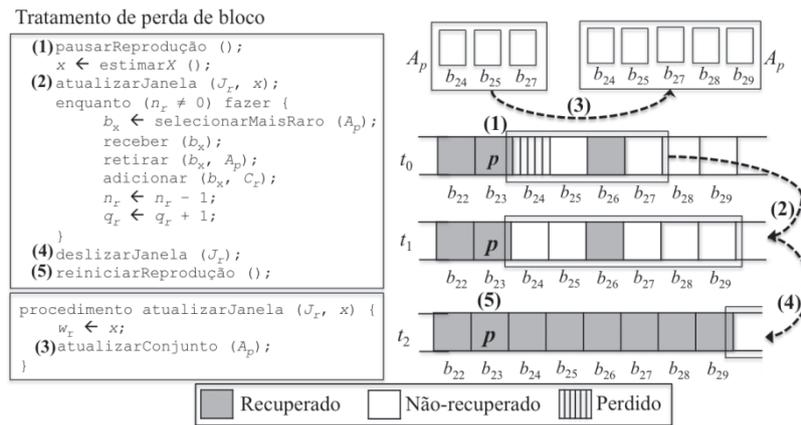
Por outro lado, a janela de previsão é atualizada. O passo (2) demonstra que o modelo de comportamento do usuário retorna um bloco a partir do salto no ponto de reprodução realizado. De acordo com o passo (3), o modelo de comportamento retornou o bloco  $b_{19}$ , e a janela de previsão deslizou até englobar o bloco  $b_{20}$ , ou seja, o primeiro bloco não recuperado a partir desse ponto. Por fim, o passo (4) atualiza o conjunto  $C_p$ .

Quando um bloco não é recuperado a tempo de ser reproduzido, o arquivo multimídia deve ser pausado e uma quantidade  $x$  de blocos deve ser estimada para determinar o tamanho da janela  $J_r$ . Essa estimativa baseia-se na média de blocos consecutivos visualizados pelo usuário antes de serem executadas ações interativas por ele. Um

usuário que executa poucas ações interativas possui um  $x$  maior que outro que executa muitas ações interativas.

Essa situação é demonstrada na Figura 10. Conforme o passo (1), a reprodução é pausada enquanto o bloco  $b_{23}$  estava sendo reproduzido porque o bloco  $b_{24}$  foi perdido. Ressalta-se que, nesse momento,  $w_r = w_{r,min}$ . Em seguida, o valor de  $x$  é estimado e, no passo (2), a janela  $J_r$  atualiza o seu tamanho  $w_r$  de acordo com  $x$ . No exemplo, a janela  $J_r$ , que englobava quatro blocos em seu interior, no instante  $t_0$  passa a englobar seis blocos no instante  $t_1$ . Por sua vez, o conjunto  $A_p$  é atualizado no passo (3) e, após todos os blocos desse conjunto serem recebidos, a janela desliza até o próximo bloco não recuperado pelo usuário, conforme o passo (4).

Figura 10 - Exemplo e algoritmo para o tratamento de perda de bloco do BISI



Fonte: do Autor

Finalmente, no passo (5), a reprodução reinicia a partir do ponto em que havia parado. Entende-se que esse tratamento de perda de blocos é importante porque previne que usuários com baixo desempenho no *download* tenham uma quantidade excessiva de interrupções. Isso é realizado levando-se em consideração o nível de interatividade de cada usuário.

#### 4.2.2 Tabela comparativa

A Tabela 2 compara as características dos principais protocolos baseados no BitTorrent e voltados para *streaming* em ambientes interativos. Todos os protocolos apresentados na Tabela 2 não alteram a política de seleção de *peers* do BitTorrent. Além disso, em todos os protocolos, utilizam-se duas janelas para a seleção de blocos: janela de reprodução e de previsão.

**Tabela 2** - Tabela Comparativa com os principais protocolos baseados no BitTorrent para *streaming* de objetos multimídia em ambientes interativos

Protocolo	Política de Seleção de Peers	Política de Seleção de Blocos	Tamanho da Janela de Reprodução	Tamanho da Janela de Previsão	Local de Seleção de Blocos
BIP-F	BitTorrent original	Janela Dupla Simples	Fixo	Fixo	Interior e exterior das janelas
BIP-S	BitTorrent original	Janela Dupla Simples	Fixo	Fixo	Interior das janelas
BISI	BitTorrent original	Janela Dupla Limitada	Dinâmico	Fixo	Interior das janelas

Fonte: do Autor

Nos protocolos BIP-F e BIP-S, as janelas possuem uma quantidade fixa de blocos em seu interior durante o *download* de todo o arquivo. O protocolo BISI define que as janelas, por outro lado, possuem comportamentos diferentes. O tamanho da janela de reprodução adapta-se dinamicamente de acordo com o progresso de *download* do usuário e a janela de previsão mantém um tamanho fixo.

Além disso, a utilização de janelas não necessariamente restringe os blocos que devem ser recuperados. No caso do protocolo BIP-F, a seleção ocorre tanto dentro quanto fora das janelas. Assim, proporciona-se maior diversidade de blocos ao sistema e, ao mesmo tempo, não se restringe a recuperação apenas daqueles com maior prioridade. Por outro lado, nos protocolos BIP-S e BISI, os blocos são recuperados apenas dentro das duas janelas de forma alternada.

O protocolo BISI considera duas questões importantes. Em primeiro lugar, possibilita a variação do tamanho da janela de reprodução apenas quando o usuário possui um bom desempenho. Dessa forma, proporciona uma maior diversidade de blocos que o protocolo BIP-S sem comprometer o *download* do usuário como no protocolo BIP-F. Em segundo lugar, limita a recuperação de blocos no interior da janela. Assim, ao contrário do protocolo BIP-F, apenas os blocos que provavelmente serão reproduzidos em breve são recuperados. Isso torna o protocolo BISI potencialmente mais eficiente.

## 5 Simuladores

### 5.1 Fundamentos

De acordo com Bhardwaj, Upadhyay e Dixit (2010), simulação é o processo de modelagem de um sis-

tema com o objetivo de estudá-lo cientificamente. Nesse sentido, entende-se que simulador é o ambiente que suporta a realização de simulações, ou seja, a modelagem e a representação de sistemas.

A utilização de simuladores em vez de experimentos em redes reais na análise de aplicações P2P justifica-se principalmente pelos custos reduzidos na implementação e na avaliação de modelos de simulação (BHARDWAJ; UPADHYAY; DIXIT, 2010). Além disso, as simulações são realizadas em ambientes controlados, que facilitam a avaliação de diferentes parâmetros.

Porém, a simulação de redes P2P, que possuem uma natureza dinâmica e milhares de usuários (nós) simultâneos, ainda assim pode ser custosa e exigir muitos recursos computacionais. Daí, com o objetivo de reduzir esses custos, os simuladores podem acabar não considerando todos os aspectos e elementos presentes nas redes reais.

Os simuladores de redes são classificados em três categorias diferentes em função do seu grau de abstração: (i) simuladores baseados em pacotes; (ii) simuladores baseados em fluidos; e (iii) simuladores híbridos.

Os simuladores baseados em pacotes funcionam no nível de rede e, assim, todo pacote transmitido pelos nós da rede é tratado de forma individual. Os simuladores baseados em fluidos funcionam no nível de aplicação e permitem uma maior abstração dos componentes. Nesse caso, representa-se o tráfego da rede como um fluxo de fluido contínuo, ao invés de uma sequência discreta de pacotes (AGUIAR et al., 2008). Por fim, os simuladores híbridos foram desenvolvidos para simulações mais abrangentes, permitindo que os usuários determinem se a simulação realizada será baseada em pacotes ou em

fluidos. Dessa forma, essa categoria fornece uma maior flexibilidade e permite que o usuário defina o nível de detalhamento desejado para a simulação de acordo com a sua necessidade.

Em comparação com os simuladores baseados em fluidos, os simuladores baseados em pacotes possibilitam a modelagem de elementos da rede com um alto grau de precisão. Por outro lado, esses simuladores são mais complexos e a escalabilidade do modelo a ser simulado tende a ser menor e dependente de características como a quantidade de memória disponível e a velocidade do processamento computacional.

Para a avaliação de modelos do BitTorrent, os simuladores baseados em pacotes exigem que toda a camada de rede seja modelada. Kangasharju et al. (2007) argumentam que esses detalhes são irrelevantes quando o objetivo principal da simulação é o teste de diferentes estratégias de seleção de blocos e de *peers*. Por outro lado, Eger et al. (2009) defendem que os detalhes da camada de rede são significativos e determinam o *delay* (atraso) de propagação dos dados entre os *peers*, influenciando na performance do *download* e na política de seleção de *peers* do BitTorrent.

Os simuladores baseados em pacotes, de fato, permitem a análise de resultados mais próximos dos obtidos em sistemas reais. Porém, nem sempre esse é o objetivo principal da simulação. Para as análises competitivas de diferentes estratégias de seleção de blocos do BitTorrent, por exemplo, o objetivo não é obter os resultados reais, mas sim concluir qual é a estratégia mais eficiente no mesmo cenário simulado. Nesse tipo de análise, a simulação baseada em fluidos é a mais adequada, pois o processo de modelagem é menos complexo e facilita o alcance dos objetivos desejados para a simulação.

As três subseções a seguir têm a finalidade de apresentar informações acerca dos principais simuladores disponíveis para cenários que envolvem protocolos P2P, com destaque para aqueles baseados no BitTorrent. Essas informações permitem escolher o simulador mais pertinente para a análise que eventualmente se deseja realizar em um determinado estudo.

## 5.2 Simuladores baseados em pacotes

O Network Simulator 2 (NS-2) (BHARDWAJ; UPADHYAY; DIXIT, 2010; XU et al., 2011) é um simula-

dor de redes bastante popular na comunidade científica, possuindo uma boa documentação sobre o seu funcionamento. Esse simulador utiliza duas linguagens de programação diferentes. Para programar o comportamento dos módulos responsáveis pela execução de protocolos e de aplicações, utiliza-se da linguagem C++. Por outro lado, a configuração do cenário a ser simulado é realizada por meio da linguagem de script OTcl.

O simulador PlanetSim (BHARDWAJ; UPADHYAY; DIXIT, 2010; XU et al., 2011), por sua vez, utiliza a linguagem de programação Java. Nele, a simulação é dividida em duas seções principais: redes sobrepostas (*overlay*) e aplicações. As redes sobrepostas são aquelas criadas na camada de aplicação independentemente da topologia da rede física. Assim, os desenvolvedores podem testar diferentes redes sobrepostas com aplicações distintas de forma simples. Porém, como desvantagem, o PlanetSim não possui um mecanismo para coletas estatísticas.

O ambiente de simulação OverSim (BHARDWAJ; UPADHYAY; DIXIT, 2010; XU et al., 2011) é voltado para a simulação de redes P2P. Esse simulador é baseado em outro mais abrangente conhecido como OMNeT++ (BHARDWAJ; UPADHYAY; DIXIT, 2010), que é apresentado de forma mais detalhada na seção de simuladores híbridos. O OverSim adiciona módulos especializados ao OMNeT++ com o objetivo de facilitar a modelagem e a avaliação de sistemas P2P.

O TorrentSim, apresentado por Mansilha, Barcellos e Brasileiro (2008), é um simulador desenvolvido para avaliações do protocolo BitTorrent que utiliza a linguagem de programação Java. Ele está inserido em um ambiente integrado para avaliação de redes BitTorrent denominado TorrentLab.

O TorrentLab possui um ambiente de modelagem utilizado pelo usuário para descrever a rede BitTorrent a ser simulada. Também possui o TorrentSim, conforme já mencionado, responsável pelas simulações, e o TorrentExp, responsável pelos experimentos na rede do modelo especificado. Por fim, as saídas produzidas pelo TorrentSim e pelo TorrentExp são avaliadas em um ambiente de consolidação de resultados.

## 5.3 Simuladores baseados em fluidos

O simulador ChunkSim (KANGASHARJU et al., 2007) é escrito em Java e é desenvolvido especificamente

para a avaliação de sistemas de distribuição de conteúdo P2P. Esse simulador busca facilitar a modificação e a extensão de diferentes estratégias de seleção de blocos e de *peers*.

O ambiente de simulação General P2P Simulator (GPS) (BHARDWAJ; UPADHYAY; DIXIT, 2010; XU et al., 2011) é escrito em Java e especializado em redes P2P. É caracterizado por ser mais preciso que outros simuladores baseados em fluidos, pois permite que algumas funcionalidades da rede sejam modeladas. As principais implementações realizadas nesse simulador são voltadas para a avaliação do protocolo BitTorrent e, por isso, pouco suporte é fornecido para a implementação de outros protocolos.

Por sua vez, o simulador OctoSim (BHARAMBE; HERLEY; PADMANABHAN, 2006) é escrito em C# e desenvolvido exclusivamente para a avaliação do protocolo BitTorrent. Nele é possível modelar as atividades de cada *peer* e os principais mecanismos do BitTorrent (seleção de blocos e de *peers*) de forma detalhada.

Além disso, alguns simuladores baseados em fluidos foram desenvolvidos especificamente para a avaliação de aplicações *streaming*. O Overlay P2P Streaming Simulator (OPSS) (XU et al., 2011), por exemplo, é designado para a simulação de sistemas P2P *streaming*. Ele utiliza a linguagem de programação C++ e possui uma arquitetura baseada em três camadas básicas: usuário, *overlay* e rede.

A camada usuário representa o comportamento dos *peers*. A camada *overlay* representa o mecanismo de *streaming* a ser utilizado, incluindo, por exemplo, o controle de comunicação entre os nós. Por fim, na camada rede, como o próprio nome sugere, representa-se o comportamento da rede.

O simulador apresentado por Hwang, Misra e Rubenstein (2008) realiza modificações no OctoSim para que ele passe a suportar protocolos de distribuição de conteúdos *streaming*. Nessa modificação, torna-se possível modelar o BitTorrent para suportar *streaming* de conteúdos ao agregar funcionalidades, como o controle do progresso de *download* do arquivo e o estabelecimento do bloco que deve ser selecionado em determinado instante.

#### 5.4 Simuladores híbridos

No OMNeT++ (BHARDWAJ; UPADHYAY; DIXIT, 2010), o modelo a ser simulado pelo usuário é desen-

volvido por meio da combinação de módulos simples e de módulos compostos organizados de forma hierárquica e que se comunicam entre si. O comportamento de módulos simples é programado com a linguagem C++ e, dos módulos compostos, utilizando-se a linguagem de script NED.

O simulador DEUS (XU et al., 2011) também é bastante abrangente e permite a avaliação de sistemas complexos, dentre eles, dos sistemas P2P. Basicamente, utiliza-se a linguagem Java para a implementação de nós (partes que interagem no sistema), de eventos (pedaços de código a serem agendados pelo sistema) e de processos (que regulam a ocorrência dos eventos).

Por último, menciona-se o Tangram (SILVA; LEÃO, 2009), um ambiente para a modelagem e a experimentação de sistemas computacionais. Primeiro, o usuário deve especificar um modelo. Depois, deve optar por resolver o modelo analiticamente ou via simulação, seguindo as restrições de cada método.

O método analítico consiste na solução de modelos representados por um conjunto de equações matemáticas e o método de simulação consiste na execução de modelos que normalmente são representados por uma linguagem de programação (CHWIF, 1999). Mais especificamente, no método de simulação do Tangram, utiliza-se a linguagem C++ e os testes podem ser realizados com base em pacotes ou com base em fluidos.

## 6 Considerações finais

Este trabalho apresentou dois novos protocolos P2P para *streaming* de objetos multimídia: o BitTorrent Streaming Sequencial (BISS) e o BitTorrent Streaming Interativo (BISI). Ambos são baseados no protocolo BitTorrent e utilizam o conceito de janelas a fim de restringir o conjunto de blocos de dados a serem recuperados pelo usuário.

Além disso, foram revisados os principais simuladores da literatura que podem ser utilizados para a avaliação de protocolos P2P, especialmente aqueles baseados no BitTorrent. Os simuladores encontrados foram classificados em três categorias diferentes: (i) simuladores baseados em pacotes; (ii) simuladores baseados em fluidos; e (iii) simuladores híbridos.

Dentre os principais resultados alcançados neste trabalho, são destacados os dois seguintes:

As análises e comparações realizadas permitem afirmar que os protocolos BISS e BISI são potencialmente mais eficientes do que outros protocolos da literatura, pois permitem um compromisso mais satisfatório entre os requisitos de tempo e a diversidade de blocos do sistema;

Os simuladores baseados em fluidos e os simuladores híbridos são os mais adequados para serem utilizados em análises competitivas entre protocolos P2P, especialmente quando são consideradas as estratégias de seleção de blocos e de *peers* de protocolos baseados no BitTorrent, pois ambos permitem a realização de simulações baseadas em fluidos.

Para trabalhos futuros visualiza-se analisar diferentes estratégias de seleção de *peers*, já que as pesquisas nessa área ainda são incipientes em comparação com as pesquisas na área de seleção de blocos, Também visualiza-se avaliar experimentalmente os protocolos BISS e BISI com o objetivo de ratificar os resultados encontrados nas análises conduzidas neste trabalho.

### Referências

- AGUIAR, A. S. C. et al. Análise comparativa de simuladores de redes baseados em pacotes versus simuladores utilizando abstração de fluidos. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 26., 2008, Rio de Janeiro. *Anais...* Rio de Janeiro: URFJ, maio, 2008. 177-188.
- BHARDWAJ, R.; UPADHYAY, A. K.; DIXIT, V. S. An overview on tools for peer to peer network simulation. *International Journal of Computer Applications*, EUA, v. 1, n. 19, p. 70-76, 2010.
- BHARAMBE, A. R.; HERLEY, C.; PADMANABHAN, V. N. Analyzing and improving a bittorrent network's performance mechanisms. In: INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS - INFOCOM, 25th., 2006, Barcelona. *Anais...* Barcelona, 2006.
- BORGHOL, Y. et al. A. Toward efficient on-demand streaming with bittorrent. In: INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING: networking, 2010, Chennai, India. *Anais...* Heidelberg, Berlim: Springer-Verlag, 2010.
- CARLSSON, N.; EAGER, D. L. Peer-assisted on-demand streaming of stored media using bitTorrent-like protocols. In: INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING: networking, 2007, EUA. *Anais...* EUA, 2007. 570-581.
- CHWIE, L. *Redução de modelos de simulação de eventos discretos na sua concepção: uma abordagem causal*. São Paulo. 1999. 139. Tese (Doutorado em Engenharia Mecânica) - Escola Politécnica, Universidade de São Paulo, São Paulo, 1999.
- COHEN, B. Incentives build robustness in bittorrent. In: WORKSHOP ON ECONOMICS OF PEER-TO-PEER SYSTEMS, 1., 2003, Berkeley, CA, EUA. *Anais...* Berkeley: Bittorrent.org, 22 maio 2003. p. 1-5.
- EGER, K. et al. Packet and flow level simulations of bittorrent-like p2p networks. *Multiagent and grid systems*, Amsterdam, v. 5, p. 217-232, 2009.
- HOFFMANN, L. J.; RODRIGUES, C. K. S.; LEÃO, R. M. M. BitTorrent-like protocols for interactive access to VoD systems. *European Journal of Scientific Research*, Victoria, v. 58, n. 4, p. 550-569, 2011.
- HUANG, Y. et al. Challenges, design and analysis of a large-scale P2P VoD system. In: ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION, 2008, Seattle, EUA. *Conference on Data Communication*. New York: ACM Magazines and Online Publications, 08 oct 2008. p. 375-388.
- HWANG, K. W.; MISRA, V.; RUBENSTEIN, D. *Stored media streaming in bittorrent-like P2P networks: tech report*. New York: Columbia University, (cucs-024-8) 2008.
- KANGASHARJU, J. et al. Chunksim: Simulating peer-to-peer content distribution. In: SPRING SIMULATION MULTICONFERENCE, 2007, San Diego, EUA. *Anais...* San Diego, 2007.
- LEGOUT, A.; URVOY-KELLER, G.; MICHARDI, P. Rarest first and choke algorithms are enough. In: ACM SPECIAL INTEREST GROUP ON DATA COMMUNICATION, 6., 2006, Rio de Janeiro. *Conference on Internet Measurement*, New York: ACM Magazines and Online Publications, 2006, p. 203-216.
- 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, 26., 2008, Rio de Janeiro. *Anais...* Rio de Janeiro, 2008.

SAVOLAINEN, P.; RAATIKAINEN, N.; TARKOME, S. Windowing bittorrent for video-on-demand: not all is lost with Tit-for-Tat. In: IEEE INTERNATIONAL PERFORMANCE, COMPUTING, AND COMMUNICATIONS CONFERENCE GLOBECOM, 2008, New Orleans. *Anais...* New Orleans, 2008.

SHAH, P.; PÂRIS, J. -F. Peer-to-peer multimedia streaming using bittorrent. In: IEEE INTERNATIONAL PERFORMANCE, COMPUTING, AND COMMUNICATIONS CONFERENCE, 2007, New Orleans. *Anais...* New Orleans: IEEE, 2007. p. 340-347.

SILVA, E. S.; LEÃO, R. M. M. The TANGRAM-II Integrated modeling environment for computer systems and networks. *ACM SIG-METRICS: performance evaluation review*, New York, v. 36, n. 4, 2009.

VIELMOND, C. C. L. B.; LEÃO, R. M. M.; SILVA, E. S. Um modelo HMM hierárquico para usuários interativos acessando um servidor multimedia. In: SIMPÓSIO BRASILEIRO DE REDES DE COMPUTADORES E SISTEMAS DISTRIBUÍDOS, 26., 2007, Belém. *Anais...* Belém, 2007.

VLAVIANOS, A.; ILIOFOTOU, M.; FALOUTSOS, M. BiToS: enhancing bittorrent for supporting streaming applications. In: IEEE GLOBAL INTERNET SYMPOSIUM, 9., 2006, Barcelona. *Anais...* Barcelona, 2006.

XU, H. et al. A Survey of peer-to-peer simulators and simulation technology. *Journal Of Convergence Information Technology*, Washington, p. 260-272, maio, 2011.

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