

# DATA DISPERSION AS A CRITERION FOR THE PEER-SELECTION POLICY IN A BITTORRENT NETWORK FOR INTERACTIVE ON-DEMAND STREAMING NETWORK

## A DISPERSÃO DE DADOS COMO CRITÉRIO PARA A POLÍTICA DE SELEÇÃO DE PEERS EM UMA REDE BITTORRENT PARA STREAMING SOB DEMANDA INTERATIVO

Carlo Kleber da Silva Rodrigues, Marcus Vinicius De Melo Rocha

Centro Universitário de Brasília, Assembléia do Estado de Minas Gerais

[carlokleber@gmail.com](mailto:carlokleber@gmail.com), [mvrocha@gmail.com](mailto:mvrocha@gmail.com)

**Abstract:** This article analyzes the impact of data dispersion on the peer-selection policy of a BitTorrent network used for interactive on-demand streaming service. To this end, the original peer selection policy of the BitTorrent protocol is modified so as to prioritize those peers that least introduce data dispersion into the system. Then, through simulations in real scenarios of multimedia content distribution, the system's quality of service is evaluated using different performance metrics. Compared to the traditional system, the final results allow us to conjecture that data dispersion is an important criterion for peer selection due to the service optimization that may be achieved. For example, it was possible to register reductions of up to 34% with respect to the number of data pieces then lost by clients during multimedia data transfer sessions. The main contribution of this paper is thus the possibility of implementing more efficient on-demand streaming protocols using the data-dispersion criterion.

**Keywords:** Data dispersion; Peer-selection policy; Interactive streaming; BitTorrent.

**Resumo:** Este artigo analisa o impacto da dispersão de dados na política de seleção de peers de uma rede BitTorrent utilizada para o serviço de streaming sob demanda interativo. Para tanto, a política original de seleção de peers do protocolo BitTorrent é modificada com o intuito de priorizar aqueles peers que menos introduzem dispersão. Em seguida, através de simulações em cenários reais de distribuição de conteúdo multimídia, a qualidade de serviço do sistema é avaliada usando diferentes métricas de desempenho. Comparado ao sistema tradicional, os resultados finais permitem conjecturar que a dispersão de dados é um importante critério a ser considerado para a seleção de peers em virtude da otimização de serviço que foi possível observar. Por exemplo, houve registro de reduções de até 34% com relação ao número de pedaços perdidos pelos clientes durante sessões de transferências de dados multimídia. Ante o exposto, a principal contribuição deste artigo é a possibilidade de implementação de protocolos de streaming sob demanda mais eficientes usando o critério da dispersão de dados.

**Palavras-chave:** Dispersão de dados; Política de seleção de peers; Streaming interativo; BitTorrent.

### I. INTRODUÇÃO

O protocolo BitTorrent [1][2] tornou-se um paradigma *peer-to-peer* (P2P) para a replicação de objetos na Internet, suplantando arquiteturas tradicionais baseadas no modelo cliente/servidor [3][4][5]. Dividir o objeto em pedaços e motivar os *peers* a participar no processo de replicação são os conceitos-chave desse paradigma [6][7][8].

Políticas eficientes para seleção de *peers* [9][10] e seleção de pedaços formam o núcleo do BitTorrent [11][3]. A primeira determina quais os *peers* que devem ser autorizados a receber os pedaços que outro *peer* possui, e a segunda seleciona os pedaços que devem ser solicitados por um *peer*. O sucesso desse protocolo motivou pesquisadores a adaptá-lo para *streaming* sob demanda interativo. Essa adaptação se baseia em modificar uma ou ambas de suas políticas de seleção [12][13].

As propostas que modificam a política de seleção de pedaços estabelecem um compromisso entre satisfazer os requisitos de tempo de *streaming* sob demanda e manter uma alta diversidade de pedaços [14][15][16]. As propostas que modificam a política de seleção de *peers* focam principalmente em fornecer incentivos para que os *peers* participem do processo de replicação do objeto [9][15][16].

Embora existam trabalhos anteriores para a adaptação mencionada, conforme aqueles apresentados e discutidos em [3], [14] e [17], um conceito promissor foi recentemente introduzido por Rocha e Rodrigues [18]. Conjectura-se que a política de seleção de *peers* deve levar em conta a dispersão de dados que é eventualmente introduzida devido à seleção de um determinado *peer*. Os autores fazem uma extensa discussão considerando três métricas distintas: dispersão temporal; dispersão espacial; e popularidade da posição [18].

O contexto acima fornece a motivação para este trabalho, cujo objetivo é fazer uma análise do impacto da dispersão de dados na política de seleção de *peers* em uma rede BitTorrent, usada para *streaming* sob demanda interativo. Para esse fim, a política original de seleção de *peers* do protocolo BitTorrent é modificada para priorizar *peers* que menos introduzem dispersão de dados no sistema. Em seguida, por meio de simulações em cenários de distribuição de conteúdo multimídia, avalia-se a qualidade de serviço (QoS) final do sistema, a qual se reflete principalmente pela continuidade de serviço percebida pelo usuário do sistema [3][19]. O ineditismo deste trabalho ocorre por não haver, salvo melhor juízo, outro trabalho na literatura que já tenha feito semelhante quantificação da influência da dispersão de dados.

Comparativamente com o sistema baseado na política de seleção de *peers* original do BitTorrent, os resultados aqui obtidos permitem conjecturar que a dispersão de dados é um importante critério para a seleção de *peers*. Por exemplo, é possível registrar reduções de até 34% com relação ao número de pedaços perdidos em sessões dos clientes do sistema.

O restante deste artigo está organizado como segue. A

Seção II explica o protocolo BitTorrent. Na Seção III, trabalhos relacionados são examinados. Os conceitos de dispersão e popularidade são explorados na Seção IV. A Seção V apresenta a política modificada para a seleção de *peers*. A avaliação de desempenho constitui a Seção VI. Por fim, a Seção VII traz as conclusões finais e possibilidades para pesquisas futuras.

## II. BACKGROUND: PROTOCOLO BITTORRENT

Os objetos distribuídos pelo BitTorrent são primeiro divididos em pedaços. Cada pedaço é então dividido em blocos. Os pedaços são geralmente de 256,0 kB de tamanho, enquanto os blocos têm 16,0 kB de tamanho. Apesar do fato de os blocos serem a unidade de dados na rede física, a análise do processo de replicação considera os pedaços sendo transferidos [1][2][10].

Um *leecher* é um *peer* que ainda está baixando pedaços de um objeto, mas também permite que outros *peers* baixem conteúdo dele. Um *seed* é um *peer* que já tem todas os pedaços do objeto, mas permanece no sistema apenas para permitir que outros *peers* baixem pedaços que ele possui [1][2][10].

Cada *peer* possui um conjunto de vizinhos que consiste em outros *peers* para os quais e a partir dos quais esse *peer* pode enviar e receber pedaços. O conjunto de vizinhos é definido aleatoriamente considerando uma lista *L* de *peers* que já estão baixando pedaços do objeto. Essa lista *L* é informada ao *peer* por uma entidade central denominada de *tracker*. O intercâmbio de pedaços entre os *peers* é governado pelas políticas de seleção de *peers* e de seleção de pedaços [1][2][10].

A política de seleção de *peers*, também denotada como o algoritmo de *choke*, permite que um *peer* *P* escolha quais outros *peers*, pertencentes ao seu conjunto de vizinhos, podem receber os pedaços que ele possui. Os três vizinhos que fornecem pedaços para o *peer* *P* com as mais altas taxas de dados são os escolhidos. Um *slot* de *upload* de dados é atribuído a cada um deles pelo *peer* *P* [1][2][10].

Os *peers* escolhidos são ditos no estado de *unchoked*, e os demais restantes do conjunto de vizinhos são ditos no estado de *choked*. Essa avaliação é repetida a cada 10,0 segundos, sob o que se denomina *regular unchoking*. Há também o denominado *optimistic unchoking*. Nesse caso, a cada 30,0 segundos, um *peer* escolhe aleatoriamente outro *peer* de seu conjunto de vizinhos para receber os pedaços que ele possui. Um *slot* de *upload* é alocado a esse *peer* escolhido [1][2][10].

Por sua vez, a política de seleção de pedaços é usada para escolher os pedaços que um *peer* deve solicitar. Cada *peer* tem um conjunto de pedaços mais raros. i.e., os pedaços menos replicados dentro de seu conjunto de vizinhos. Logo após passar para o estado *unchoked*, o *peer* solicita o próximo pedaço considerando seu conjunto de pedaços mais raros e os pedaços disponíveis no vizinho que acabou de passá-lo para o estado *unchoked*. Após receber o pedaço, o *peer* avisa a todos os seus vizinhos sobre o pedaço recebido [1][2][10].

## III. TRABALHOS RELACIONADOS

Esta seção apresenta trabalhos relevantes com respeito às

propostas de adaptação do protocolo BitTorrent para *streaming* sob demanda. Opta-se por uma sequência de citação que busca enaltecer o aspecto conceitual inovador introduzido pela proposta, em vez de seus detalhes de implementação. Assim, crê-se que o leitor pode mais facilmente ter uma visão contemplativa da elaboração e da diferenciação entre as propostas, bem como ter uma visualização mais objetiva do atual estado da arte.

Shah e Pâris [20] propõem realizar o *optimistic unchoke* com mais frequência. Para tanto, é definida uma janela deslizante [9] que contém os próximos *w* pedaços a serem consumidos por um cliente. No início de cada reprodução da janela, os *peers* executam um novo *optimistic unchoke*.

A proposta de Mol et al. [21] difere daquela do BitTorrent original por empregar a reciprocidade indireta. Mais precisamente, para passar o vizinho para o estado de *unchoked*, um *peer* considera a taxa de dados entre seus vizinhos, além das taxas de dados entre seus vizinhos e ele próprio.

Borghol et al. [17] definem uma janela adaptativa, cujo tamanho muda em função da quantidade de dados em ordem disponíveis no *buffer* do *peer* e da posição de reprodução atual do objeto que está sendo visualizado. Sob esta mesma ideia, Streit e Rodrigues [22] também usam uma janela cujo tamanho varia conforme o estado do sistema.

Hoffmann, Rodrigues e Leão [6] definem duas janelas de dados. Uma delas contém os pedaços de alta prioridade, e a outra contém os pedaços previstos por um modelo de comportamento do usuário. A recuperação de pedaços dentro de ambas as janelas segue a estratégia do mais raro.

D'Acuneto et al. [14] propõem três esquemas dedicados a ambientes heterogêneos, onde os *peers* podem ter diferentes capacidades de *upload* e *download*. O ajuste do número de *slots* de *upload* que um *peer* pode abrir é o conceito base de projeto desses esquemas. Neste mesmo contexto, Streit e Rodrigues [23] propõem cálculos *on-line* para ajustar o número de *slots* visando a beneficiar *peers* que não conseguem ter uma taxa de *download* satisfatória.

Rodrigues [10] realiza uma análise extensa sobre o impacto na QoS do sistema devido ao número de *slots* utilizados para *regular unchoking* e *optimistic unchoking*. Neste mesmo contexto, Rodrigues [3] propõe a utilização de uma janela deslizante concomitante com uma relação otimizada entre o número de *slots* para regular *unchoking* e para *optimistic unchoking*.

Rocha e Rodrigues [18] discutem teoricamente a dispersão de dados que é introduzida no sistema ao considerar-se uma seleção aleatória de *peers* a partir de um conjunto de vizinhos informado pelo *tracker*.

Por fim, conforme pode ser constatado pelo exposto, não há trabalhos na literatura que realizem a implementação prática do conceito da dispersão de dados na política de seleção de *peers*. Tampouco há quantificação do efeito dessa dispersão na qualidade de serviço final do sistema. É exatamente isso que é considerado na avaliação proposta neste artigo.

#### IV. DISPERSÃO

Existem dois tipos de dispersão de dados: dispersão temporal e dispersão espacial. A dispersão temporal é apenas outra maneira de indicar a taxa de solicitação de dados do cliente, indicada por  $N$ . A baixa dispersão temporal implica altas taxas de solicitação e, conseqüentemente, mais conteúdo disponível para compartilhamento. Já a alta dispersão temporal implica baixas taxas de solicitação e, conseqüentemente, menos conteúdo disponível para compartilhamento [18].

A dispersão espacial se refere à quantidade de conteúdo que dois pedidos de clientes distintos consecutivos têm em comum. Contudo que as solicitações tenham conteúdo em comum, esse conteúdo pode ser potencialmente compartilhado. Se esta sobreposição de pedidos consecutivos aumenta, então a dispersão espacial diminui. Por outro lado, se esta sobreposição de pedidos consecutivos diminui, então a dispersão espacial aumenta. Medir a dispersão temporal é direto, pois esta é inversamente proporcional à taxa de solicitação  $N$ . Por sua vez, a dispersão espacial é avaliada conforme segue.

Seja  $Q_p$  o número de vezes que uma posição  $p$  de um arquivo é solicitada por todos os clientes até o instante de tempo corrente. Assim,  $Q_p$  é a popularidade da posição  $p$  do arquivo. Qualquer posição  $p$ , recebida por um peer, está potencialmente disponível para ser compartilhada com outros peers. Sendo  $T$  o número total de posições distintas constituintes do arquivo, o potencial de compartilhamento de conteúdo, denotado por  $P$ , é definido como segue [18].

$$P = \sum_{p=0}^{T-1} (Q_p - 1) \quad (1)$$

Agora, para medir a dispersão espacial,  $D$ , é comparada a quantidade de conteúdo que é recuperado,  $M$ , com o potencial de compartilhamento de conteúdo,  $P$ , como segue [18].

$$D = 1 - \left(\frac{P}{M}\right) \quad (2)$$

Note que  $P < M$ , e  $D$  está no intervalo  $[0, 1]$ . Por exemplo, se um objeto tem uma única posição  $p$  sendo solicitada 100 vezes, então tem-se que  $Q_p = 100$  e  $P = 99$ . Como  $M = 100$  (unidades de dados), segue-se que  $D = (1 - (99/100)) = 0,01$ .

O conceito de dispersão pode ser usado para projetar uma política de seleção de peers como segue. Como mencionado, o conjunto de vizinhos resulta de uma escolha aleatória a partir de uma lista  $L$  informada pelo tracker. Seja  $k$  o número de peers dentro do conjunto de vizinhos, e seja  $n$  o número de peers dentro da lista  $L$ , onde  $n \geq k$ . Em vez de se ter uma escolha aleatória, o conjunto de vizinhos é então definido como o conjunto de  $k$  peers resultando na dispersão mínima combinada entre todos os conjuntos possíveis de  $k$  peers escolhidos dentro da lista  $L$ . Esta escolha se reduz a um problema de otimização de complexidade de tempo assintótica,  $A$ , dada pelo número de combinação de  $n$  elementos distintos tomados  $k$  de cada vez. Mais precisamente,  $A$  pode ser calculada pela equação a seguir.

$$A = \frac{n!}{k!(n-k)!} \quad (3)$$

Veja que  $A$  cresce exponencialmente. Segue-se que esta solução é proibitiva para ser considerada na prática. No entanto, uma heurística baseada em uma estratégia gulosa [24] pode ser usada em vez disso, como explicado nas etapas sucessivas a seguir:

- 1) Um peer aleatório é escolhido entre os  $n$  peers de  $L$ ;
- 2) Cada um dos  $(n-1)$  peers restantes de  $L$  é testado individualmente para ver qual, juntamente com o primeiro peer escolhido, resulta na dispersão mínima;
- 3) Cada um dos  $(n-2)$  peers restantes é testado para ver qual, juntamente com os dois peers previamente escolhidos, resulta na dispersão mínima;
- 4) Cada um dos  $(n-3)$  peers restantes é testado para ver qual, juntamente com os três peers previamente escolhidos, resulta na dispersão mínima. E assim sucessivamente até que  $k$  peers sejam escolhidos de  $L$ .

Note que a primeira etapa leva um tempo  $O(1)$  e que cada uma das demais etapas leva um tempo  $O(n)$ . Note também que essas demais etapas ocorrerão  $(k - 1)$  vezes no total, ou seja, até que existam  $k$  peers selecionados no total. Assim, a complexidade total de tempo, designada por  $S$ , pode ser então calculada pelo somatório  $S = O(1 + (k-1).n) = O(k.n) = O(n)$ . Portanto,  $S$  cresce linearmente com  $n$  e segue-se então que esta solução é de complexidade de tempo aceitável para ser utilizada na prática. Essa complexidade de tempo é que justifica a implementação do Passo 1 do Algoritmo 1 (Protocolo PDM-I), descrito na próxima seção.

#### V. POLÍTICA MODIFICADA

Por facilidade de referência, a política modificada é doravante denominada de Protocolo de Dispersão Mínima – Iterativo (PDM-I). Esse protocolo é uma variante da política original de seleção de peers do protocolo BitTorrent. A diferença entre os dois é que, no protocolo PDM-I, a seleção dos peers que vão formar o grupo de vizinhos (vide Seção II) é feita considerando os peers da lista  $L$  que menos introduzem dispersão de dados. Na política de seleção original esse critério não é levado em consideração e a escolha se dá aleatoriamente. Essa modificação é feita para permitir que ocorram mais compartilhamentos de dados entre vizinhos, resultando em maior eficiência na distribuição dos dados. Essa maior eficiência implica em um menor tempo para o recebimento dos pedaços solicitados do objeto, propiciando uma maior continuidade do serviço de *streaming*.

Cada peer tem um registro do número de vezes que cada posição do objeto é solicitada, i.e., a popularidade da posição. Este registro é usado como base para avaliar o potencial de compartilhamento de conteúdo  $P$  usando a Equação 1 e, portanto, para avaliar a dispersão usando a Equação 2. Além desse registro, cada peer mantém um registro do conteúdo do *buffer* local. O Algoritmo 1 traz a descrição passo a passo do

protocolo PDM-I. É considerado o ponto de vista de um *peer* local que acaba de ser informado da lista  $L$ . Dizer que um *peer*  $x$  está interessado em um *peer*  $y$  significa que ele (*peer*  $y$ ) tem pedaços que o *peer*  $x$  não tem; e dizer que um *peer*  $x$  não está interessado em um *peer*  $y$  significa que ele (*peer*  $y$ ) tem apenas um subconjunto dos pedaços do *peer*  $x$ .

**Algoritmo 1: Protocolo PDM-I**

Início  
 Inicialização e definições:  
Passo 1  
 Os vizinhos do *peer* local são escolhidos a partir da lista  $L$  informada pelo *tracker*, considerando o critério da dispersão mínima da heurística descrita na Seção IV. Os vizinhos são *choked* e classificados em interessados e não interessados.  
Passo 2  
 A utilização da capacidade de banda do *peer* se dá de maneira idêntica àquela da proposta original do protocolo BitTorrent.  
 Repetir:  
Passo 3  
 A cada  $\delta = 10,0$  segundos, realizar *regular unchoking*, conforme proposta original do protocolo BitTorrent.  
Passo 4  
 A cada  $\delta = 30,0$  segundos, realizar *optimistic unchoking*, conforme proposta original do protocolo BitTorrent.  
 Até que o conteúdo desejado seja baixado.  
 Fim.

VI. AVALIAÇÃO DE DESEMPENHO

A. Caracterização de Cenário

Os cenários de simulação são concebidos a partir da análise de cargas de trabalho reais interativas de dois domínios de aplicação diferentes: acadêmico e não acadêmico.

Para o domínio acadêmico, são considerados o sistema de ensino eTeach, um servidor da Universidade de Wisconsin-Madison, tendo 46.958 pedidos de vídeos de duração de até 5 minutos e de 50-60 minutos, e o sistema Manic, um sistema de vídeo educacional da Universidade de Massachusetts, com um total de 25.833 pedidos de vídeos de durações semelhantes àquelas do sistema eTeach [18][25].

Para o domínio não acadêmico, há arquivos de áudio e vídeo, geralmente com menos de 10 minutos de duração, sendo provenientes de dois provedores de conteúdo da América Latina. O primeiro é o sistema Universo Online (UOL), com 5.385.822 pedidos para objetos de rádio *on-line*, e 1.453.117 solicitações para objetos de vídeo. O segundo possui 4.160.889 pedidos para objetos de rádio *on-line*; por razões de confidencialidade, esse servidor não é aqui identificado.

Essas cargas de trabalho são ainda agrupadas em três perfis de interatividade com base na duração da solicitação e na taxa de solicitação, dentro da mesma sessão do cliente, como segue:

- Alta interatividade (HI): solicitações curtas (menos de 20% do objeto), pelo menos 3 solicitações em uma sessão, típico de vídeos acadêmicos;
- Baixa interatividade (LI): solicitações mais longas (pelo menos 20% do objeto), menos de 2 solicitações

em uma sessão, típico de áudio não acadêmico e vídeos muito curtos;

- Interatividade média (IM): solicitações curtas (menos de 20% do objeto), menos de 3 solicitações em uma sessão, típico de vídeos não acadêmicos.

No total, existem 36 cargas de trabalho reais distintas. Considerando o perfil de interatividade, elas são distribuídas da seguinte forma: 17 cargas HI; 7 cargas MI; e 12 cargas LI. Considerando o domínio da aplicação, elas são distribuídas da seguinte forma: 7 cargas de tipo Áudio Comercial (A.C); 9 cargas de tipo Vídeo Comercial (V.C), e 20 cargas de tipo vídeo educacional (V.C). A partir da análise e tratamento dessas cargas reais, são então geradas, para cada uma das 36 cargas reais originais, 3 cargas sintéticas com 5 minutos de duração cada uma, perfazendo um total de 108 cargas sintéticas. São servidos 10, 25 e 50 clientes. Ademais, cada pedaço do arquivo tem duração de 2 segundos, tendo cada arquivo um tamanho total de 24 a 2.184 pedaços, conforme a duração da mídia. Essas cargas sintéticas são usadas como entrada para os modelos de simulação descritos mais adiante

B. Ambiente de Simulação e Métricas

Dois modelos foram desenvolvidos em um ambiente de simulação orientado por eventos. Esse ambiente foi escrito em linguagem de programação C. Foram tomados cuidados quanto à geração de números pseudoaleatórios, uma vez que os geradores nativos das bibliotecas padrão costumam ser pouco confiáveis na garantia da interindependência dos números [26][27]. Por isso, foi aqui utilizada a biblioteca SPRNG [26].

O primeiro modelo seguiu o protocolo BitTorrent original, com trocas de mensagens sendo simuladas pela geração de eventos de simulação. A partir deste modelo, criou-se o segundo modelo para representar a nova proposta: PDM-I. Esse segundo modelo considera o tempo de chegada de cada pedaço do objeto, como forma de determinar o eventual atraso na chegada de pedaços para exibição imediata, e implementa a política de seleção de *peers* baseada na dispersão.

As métricas usadas nas simulações estão na Tabela I e foram definidas com base em trabalhos anteriores da literatura, como [28], [8] e [29], priorizando observar a qualidade de serviço (QoS) final percebida pelo usuário. Essa qualidade se traduz pela continuidade do serviço, ou seja, máxima ausência de interrupções durante a exibição do objeto e tempo mínimo para retornar à exibição após uma inevitável interrupção. Medir cada uma das métricas individualmente fornece importantes evidências do nível de QoS experimentado por um cliente do sistema. No entanto, sabe-se que é a análise conjunta desses métricas que se torna mais valiosa para avaliação do protocolo PDM-I.

TABELA I  
 MÉTRICAS DE DESEMPENHO

Métrica	Notação	Definição
---------	---------	-----------

Coeficiente de Interrupção	CI	Número médio de <i>pedaços</i> perdidos durante a reprodução do arquivo do cliente normalizado pelo tamanho total do arquivo em número de <i>pedaços</i> . Numericamente, está dentro do intervalo [0, 1]. Quanto mais próximo estiver de 0, mais eficiente será o protocolo.
Tempo de Serviço Normalizado	TSN	Tempo médio necessário para um cliente começar a receber os dados solicitados normalizado pelo tamanho total do arquivo em número de <i>pedaços</i> . Usado principalmente para fornecer comparações relativas justas em termos de tempo de serviço em cenários de distribuição de diferentes tamanhos de arquivo.
Tempo de Serviço de Requisições Adicionais	TRN	Tempo médio necessário para um cliente começar a receber os dados solicitados nas requisições subsequentes àquela inicial, normalizado pelo tamanho total do arquivo em número de <i>pedaços</i> . Usado principalmente para fornecer comparações relativas justas em termos de tempo de serviço em cenários de distribuição de diferentes tamanhos de arquivo.
Coeficiente de Perda de Requisições	CP	Número médio de requisições que não puderam ser servidas porque o cliente estava ocupado atendendo a requisições anteriores, normalizado pelo número total de requisições. Numericamente, está dentro do intervalo [0, 1]. Quanto mais próximo estiver de 0, mais eficiente será o protocolo.

### C. Resultados e Análises

Os resultados aqui incluem também aqueles relacionados ao protocolo designado de Protocolo de Referência de Simulação - PRS. Esse protocolo é idêntico ao PDM-I, exceto no que se refere ao Passo 1 do Algoritmo 1. Ou seja, a política de seleção de *peers* do PRS não leva em conta a dispersão resultante. Por meio de comparação, é possível então observar a otimização obtida por evitar-se a dispersão dos dados.

A Tabela II traz então uma avaliação comparativa dos desempenhos dos protocolos PDM-I e PRS para as 108 cargas sintéticas, agrupadas na correspondente carga de trabalho real (id=1, ...,36), destacando as métricas definidas na Tabela I, o tipo de carga, e o perfil de interatividade (PI). Cada métrica da tabela recebe uma nota (sinais "+", "-" e "0"), que leva em conta as situações com 10, 25 e 50 *peers*.

Mais precisamente, os sinais "+", "-" e "0" são usados para indicar o desempenho do protocolo PDM-I frente ao protocolo PRS. O sinal "0" indica um empate entre os protocolos para as três situações. O sinal "+" indica que o protocolo PMD-I é superior ao PRS em pelo menos uma das três situações, sendo próximo ao mesmo nas restantes. Por superior, entende-se qualquer nível de otimização percebido. O sinal "-" indica o oposto de "+", ou seja, o protocolo PRS é superior ao PDM-I em pelo menos uma das três situações, sendo próximo ao mesmo nas restantes. Na avaliação da vantagem, cada sinal "-" anula um sinal "+", sendo o sinal "0" desconsiderado.

Conforme os resultados contidos na Tabela II, vê-se que o PDM-I é vantajoso em 23 situações, enquanto que o protocolo PRS é vantajoso em 13 situações. Além disso, o protocolo

PDM-I é mais vantajoso em cargas de vídeo, especialmente no caso de vídeo acadêmico, com vantagem em 17 das 20 cargas. Por outro lado, o protocolo PRS se mostra superior nas cargas de áudio, com vantagem em 6 das 7 cargas.

Uma avaliação quantitativa dos protocolos PDM-I e PRS é ainda mostrada nas duas figuras comentadas a seguir. A Fig. 1 mostra as métricas TNS e CI para uma mesma carga de vídeo acadêmico de alta interatividade (HI). Os valores observados para essas métricas evidenciam uma vantagem para o PDM-I, especialmente quanto à menor interrupção de requisições. Já a Fig. 2 mostra as métricas CP, CI e TNS para uma mesma carga de vídeo acadêmico de média interatividade (MI). Os valores obtidos para essas métricas evidenciam uma vantagem para o PDM-I, especialmente quanto à perda de pedaços e à interrupção de requisições. Assim como no caso anterior, a métrica TNS, que mede o tempo de serviço normalizado para o início da exibição, é próximo nos dois protocolos, mas com vantagem para o PDM-I.

Por fim, ante os resultados aqui apresentados, é possível concluir que o uso da dispersão, como critério para seleção de *peers*, é uma estratégia promissora para melhorar o desempenho de protocolos empregados para *streaming* sob demanda interativo. Essa conclusão deriva especialmente pela observação de que a maioria dos cenários analisados foram favoráveis ao PDM-I (Tabela II), destacando-se os menores valores obtidos pelo PDM-I para a métrica CI, quando otimizações de até 34% puderam ser contempladas com relação aos valores obtidos pelo PRS (Fig. 1 e Fig. 2).

## VII. CONCLUSÕES E TRABALHOS FUTUROS

Este artigo analisou o impacto da dispersão de dados na política de seleção de *peers* em uma rede BitTorrent usada para *streaming* sob demanda interativo. Para tanto, a política original de seleção de *peers* do BitTorrent foi modificada para priorizar *peers* que menos introduziam dispersão de dados. A mensuração desse impacto foi obtida com simulações em cenários reais de distribuição de conteúdo multimídia. Diferentes métricas de desempenho foram utilizadas para verificar a qualidade final percebida no sistema.

Comparativamente com a política tradicional de seleção de *peers* do BitTorrent, os resultados mostram-se favoráveis ao uso do conceito da dispersão de dados na maioria dos casos analisados, sendo possível alcançar otimizações de até 34% com relação ao número de pedaços perdidos em sessões dos clientes. Isso permite conjecturar sobre o promissor emprego do conceito de dispersão em cenários de diferentes naturezas de mídia (p. ex., vídeo e áudio) e de diferentes domínios de aplicação (p. ex., acadêmico e não acadêmico).

É preciso ainda destacar que a principal contribuição deste artigo foi a comprovação da possibilidade de implementação de protocolos de *streaming* sob demanda interativo mais eficientes usando o critério da dispersão de dados. Além disso, o ineditismo deste trabalho ocorreu por não haver, salvo melhor juízo, outro trabalho na literatura que já tenha feito semelhante quantificação da influência da dispersão de dados.

Por fim, como trabalhos futuros, visualizam-se os seguintes: (i) analisar a implementação conjunta do critério da dispersão com outros critérios da literatura com vistas à otimização da política de seleção de *peers* [10]; (ii) proceder de forma análoga à anterior, só que considerando critérios de seleção de pedaços, especialmente como relação emprego de janelas deslizantes [11]; (iii) analisar perfis de comportamento de usuários em plataformas reais de Educação a Distância (EaD) para retificar ou ratificar critérios para o desenvolvimento de protocolos de *streaming* específicos para EaD [30].

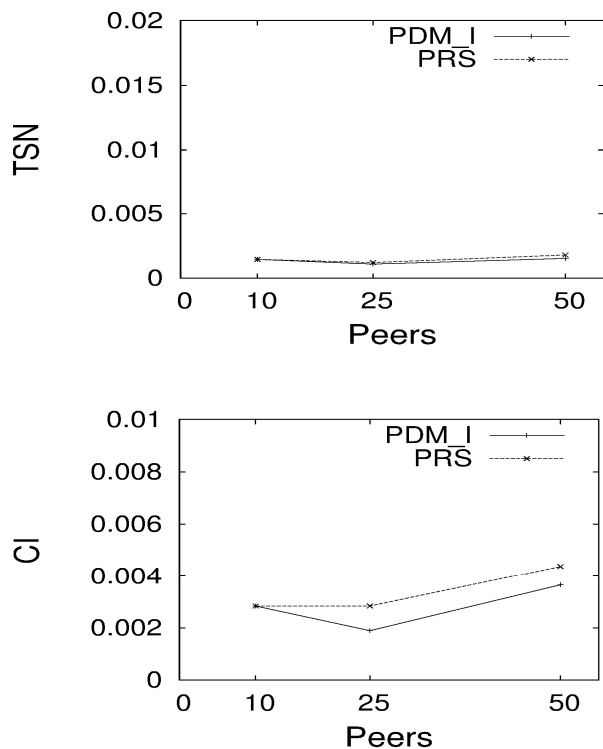
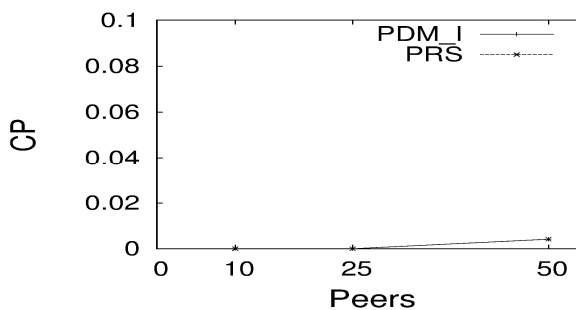


Figura 1. TSN e CI para carga HI de vídeo acadêmico.

7	+	-	+	+	MI	V.A.	PDM-I
8	0	+	+	+	HI	V.A.	PDM-I
9	0	+	+	+	HI	V.A.	PDM-I
10	+	-	+	+	HI	V.A.	PDM-I
11	-	+	+	+	HI	V.A.	PDM-I
12	0	+	+	+	HI	V.A.	PDM-I
13	0	-	+	+	HI	V.A.	PDM-I
14	0	+	+	0	HI	V.A.	PDM-I
15	0	-	-	+	HI	V.A.	PRS
16	0	-	+	+	HI	V.A.	PDM-I
17	0	+	+	0	MI	V.A.	PDM-I
18	0	+	-	+	MI	V.A.	PDM-I
19	-	+	+	+	HI	V.A.	PDM-I
20	+	+	-	+	HI	V.A.	PDM-I
21	0	-	-	-	LI	A.C.	PRS
22	-	-	-	+	LI	A.C.	PRS
23	0	-	-	+	LI	A.C.	PRS
24	+	+	+	+	HI	A.C.	PDM-I
25	+	0	-	0	LI	V.C.	PRS
26	0	-	-	-	LI	V.C.	PRS
27	0	+	-	+	LI	V.C.	PDM-I
28	0	-	+	+	LI	V.C.	PDM-I
29	0	-	-	+	MI	V.C.	PRS
30	0	+	+	+	MI	V.C.	PDM-I
31	0	+	-	+	HI	V.C.	PDM-I
32	0	-	-	-	MI	V.C.	PRS
33	-	+	+	+	MI	V.C.	PDM-I
34	0	-	-	+	LI	A.C.	PRS
35	0	-	-	0	LI	A.C.	PRS
36	0	-	-	0	LI	A.C.	PRS

TABELA II  
RESULTADOS DE SIMULAÇÃO

id	CP	CI	TSN	TRN	PI	Tipo	Superior
1	0	+	+	+	HI	V.A.	PDM-I
2	+	+	-	0	LI	V.A.	PDM-I
3	0	-	+	-	HI	V.A.	PRS
4	+	-	+	-	HI	V.A.	PRS
5	0	+	+	+	HI	V.A.	PDM-I
6	0	+	+	0	LI	V.A.	PDM-I



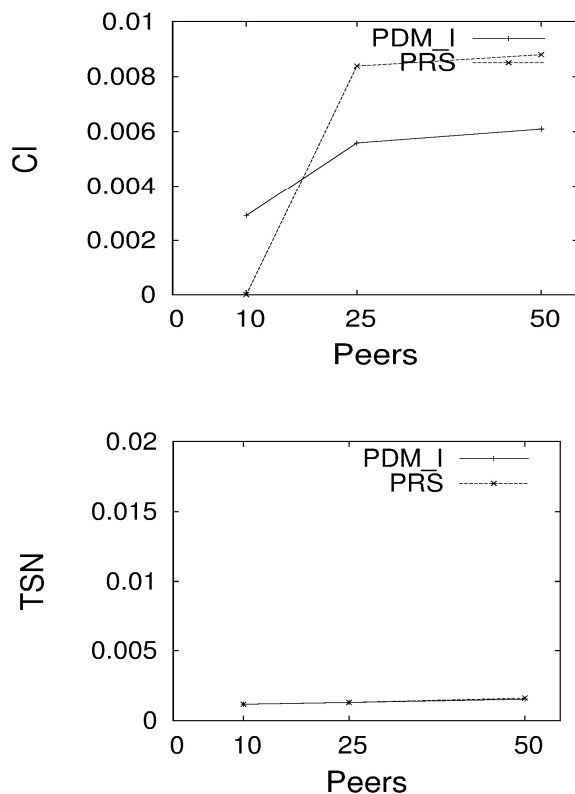


Figura 2. CP, CI e TSN para carga MI de vídeo acadêmico.

#### REFERÊNCIAS

- [1] Cohen, B. "Incentives build robustness in BitTorrent". In First Workshop on Economics of Peer-to-Peer Systems, Berkeley, USA, 2003.
- [2] Legout, A.; Urvoy-Keller, G.; Michiardi, P. "Rarest first and choke algorithms are enough". In 6th ACM SIGCOM Conference on Internet Measurement, Rio de Janeiro, RJ, Brazil, 2006.
- [3] Rodrigues, C. K. S. "On the optimization of BitTorrent-like protocols for interactive on-demand streaming systems". International Journal of Computer Networks & Communications, vol. 6, no. 5, pp. 39-58, 2014.
- [4] Rodrigues, C. K. S.; Leão, R. M. M. "Bandwidth usage distribution of multimedia servers using Patching". Computer Networks, vol. 51, no. 3, pp. 569 – 587, 2007.
- [5] Azzedin, F.; Yahaya, M. "Modeling BitTorrent choking algorithm using game theory". Future Generation Computer Systems, vol. 55, pp. 255-265, February 2016.
- [6] 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, vol. 58, no. 4, pp. 550-569, 2011.
- [7] Adamsky, F.; Khayam, S. A.; Jäger, R.; Rajarajan, M. "Stealing bandwidth from BitTorrent seeders". Computers & Security, vol 46, pp. 126-140, October 2014.
- [8] Varvello, M., Steiner, M.; Laevens, K. "Understanding BitTorrent: a reality check from the ISP's perspective". Computer Networks, vol.56, no. 40, pp. 1054-1065, 2012.
- [9] D'Acunto, L.; Andrade, J.; Sips, H. "Peer selection strategies for improved QoS in heterogeneous BitTorrent-like VoD systems". In IEEE International Symposium on Multimedia, Taichung, Taiwan, 2010.
- [10] Rodrigues, C. K. S. "Analyzing peer selection policies for BitTorrent multimedia on-demand systems in Internet". International Journal of Computer Networks & Communications, vol. 6, no. 1, pp. 203-221, 2014.
- [11] Romero, P.; Amoza, F. R.; Rodríguez-Bocca, P. "Optimum piece selection strategies for a peer-to-peer video streaming platform". Computer and Operations Research, vol. 51, no. 5, pp. 1289-1299, 2013.
- [12] Ramzan, N.; Park, H.; Izquierdo, E. "Video streaming over P2P networks: Challenges and opportunities". Signal processing: Image Communication, vol. 27, pp 401 – 411, 2012.
- [13] Hua, Kai-Lung; Chiu, Ge-Ming; Pao, Hsing-Kuo; Cheng, Yi-Chi. "An efficient scheduling algorithm for scalable video streaming over P2P networks". Computer Networks, vol. 57, no. 14, pp 2856 – 2868, 2013.
- [14] D'Acunto et al. "BitTorrent-like P2P approaches for VoD: A comparative study". Computer Networks, vol. 57, no. 5, pp. 1253-1276, 2013.
- [15] Ma, Z; Xu, K.; Liu, J.; Wang, H. "Measurement, modeling and enhancement of BitTorrent-based VoD system". Computer Networks, vol 56, no. 3, pp. 1103-1117, February 2012.
- [16] Yang, Y.; Chow, A.; Golubchik, L.; Bragg, D. "Improving QoS in BitTorrent-like VoD systems". In Proceedings of the IEEE INFOCOM, San Diego, CA, USA, 2010.
- [17] Borghol, Y. et al. "Toward efficient on-demand streaming with BitTorrent". In: IFIP Networking, Chennai, India, 2010.
- [18] Rocha, M. V. M.; Rodrigues, C. K. S. "On client's interactive behaviour to design peer selection policies for BitTorrent-like protocols". International Journal of Computer Networks & Communications (IJCNC), vol. 5, no. 5, pp. 141-159, 2013.
- [19] Carlsson, N.; Eager, D. L. "Peer-assisted on-demand streaming of stored media using BitTorrent-like protocols". In IFIP/TC6 NETWORKING, Georgia, USA, 2007.
- [20] Shah, P.; Pâris, J.-F. "Peer-to-Peer multimedia streaming using BitTorrent". In IEEE International Performance, Computing, and Communications Conference (IPCCC), New Orleans, Louisiana, USA, 2007.
- [21] Mol, J. et al. "Give-to-Get: Free-riding-resilient video-on-demand in P2P systems". In SPIE MMCN, San Jose, California, USA, 2008.
- [22] Streit, A. G.; Rodrigues, C. K. S. "Proposing a BitTorrent-like protocol for efficient interactive multimedia streaming applications". The SIJ Transactions on Computer Networks and Communications Engineering, vol. 3, no. 2, pp. 29-37, 2015.
- [23] Streit, A. G.; Rodrigues, C. K. S. "Improving BitTorrent's peer selection for multimedia content on-demand delivery". International Journal of Computer Networks & Communications, vol. 7, no. 6, pp. 111-124, 2015.
- [24] Cormen, T.H; Leiserson, C. E.; Rivest, R. L. Introduction to Algorithms. 3rd Edition. Cambridge, Massachusetts: MIT Press, 2009.
- [25] Rocha, M. et al. "Scalable media streaming to interactive users". In ACM MULTIMEDIA, Singapore, 2005.
- [26] Mascagni, M.; Srinivasan, A. "SPRNG: a scalable library for pseudorandom number generation". ACM Transactions on Mathematical Software (TOMS), vol. 26, no. 3, pp. 436-462, 2000.
- [27] Ross, S. M. Simulation. 5th Edition. San Diego, California: Elsevier, 2013.
- [28] de Souza e Silva, E., Leão, R., Menasché, D. & Rocha, A. "On the interplay between content popularity and performance in P2P systems". Lecture Notes in Computer Science (LNCS), vol. 8054, pp. 3-21, 2013.

- [29] Wang, H., Liu, J.; Xu, K. "Understand traffic locality of peer-to-peer video file swarming". *Computer Communications*, vol. 35, no. 15, pp. 1930-1937, 2012.
- [30] D. Zegarra Rodríguez, G. Bressan, "Video quality assessments on digital TV and video streaming services using objective metrics".
- [31] *IEEE Latin America Transactions*, vol. 10, no. 1, pp. 1184-1189, 2012.