

BIB-R: Uma Nova Adaptação do BitTorrent para *Streaming* de Vídeo sob Demanda ante Clientes Interativos em MANETs

Carlo K. da S. Rodrigues¹, Vladimir E. M. Rocha¹

¹Centro de Matemática, Computação e Cognição (CMCC)
Universidade Federal do ABC (UFABC) – Santo André – SP – Brasil

{carlo.kleber, vladimir.rocha}@ufabc.edu.br

Abstract. *This article proposes a new adaptation of BitTorrent for video-on-demand streaming in MANETs with interactive clients. As the main innovative aspects, this proposal makes use of broadcast traffic and assesses the length of the routes, in number of hops, to be used to disseminate the video. Through simulations and evaluating different metrics, the proposed adaptation is validated and its performance is satisfactorily proven. For instance, compared to recent proposals in the literature, the final results attest to optimizations of up to 51% and 89% at the video-retrieval rate and at the display discontinuity time, respectively, which confirms adequate levels of system QoS and client QoE.*

Resumo. *Este artigo propõe uma nova adaptação do BitTorrent para streaming de vídeo sob demanda ante clientes interativos em MANETs. Como aspectos mais inovadores, esta proposta faz uso de tráfego broadcast e avalia o comprimento das rotas, em número de saltos, a serem utilizadas para disseminação do vídeo. Por meio de simulações e avaliando-se diferentes métricas, a adaptação proposta é validada e tem seu desempenho satisfatoriamente comprovado. Por exemplo, comparativamente a propostas recentes da literatura, os resultados finais atestam otimizações de até 51% e 89% na taxa de recuperação do vídeo e no tempo de descontinuidade de exibição, respectivamente, o que evidencia adequados níveis de QoS do sistema e de QoE do cliente.*

1. Introdução

Streaming de vídeo sob demanda por meio de redes sem fio móveis tem apresentado um crescimento vertiginoso no mundo digital [CISCO 2019, Machado et al. 2019]. Por um lado, há um natural apelo pelo consumo de conteúdo multimídia em virtude da diversidade de aplicações existentes, que contemplam entretenimento, ensino, negócios, operações militares, dentre outras. Por outro lado, a tecnologia sem fio torna a distribuição desse conteúdo mais ubíqua e pervasiva, pois possibilita o estabelecimento de comunicação de forma mais flexível, abrangendo inclusive ambientes inóspitos e de difícil acesso.

Há dois tipos de redes sem fio móveis: infraestruturada e sem infraestrutura. No primeiro, há um sistema fixo e cabeado, que inclui antenas e estações bases, ao qual os nós móveis se conectam por meio de enlaces sem fio. Esse sistema provê toda a gerência da comunicação entre os nós. No segundo tipo, também conhecido pelo acrônimo MANET (*Mobile Ad Hoc Network*), os próprios nós atuam como servidores, clientes e roteadores, havendo a comunicação sem dependência de qualquer infraestrutura fixa. Essa condição

dá às MANETs as características de autoconfiguração, autoformação e automanutenção, tornando-as de especial interesse [Wang et al. 2009, Fleury and Qadri 2019].

Em resposta a esse cenário emergente, pesquisas têm sido realizadas em prol da implementação eficiente de *streaming* de conteúdo multimídia em redes sem fio móveis. Um grupo de soluções considera a adaptação do conhecido protocolo BitTorrent (e.g., [Rodrigues and Rocha 2019c, Rodrigues and Rocha 2019a]). O núcleo deste protocolo é constituído por uma política de seleção de pedaços e uma política de seleção de *peers*. Os arquivos a serem replicados são divididos em pedaços, e as mencionadas políticas controlam a transmissão desses pedaços pela rede de comunicação, determinando que pedaços devem ser solicitados com maior prioridade e, ainda, que *peers* devem ser habilitados a receber esses pedaços. Os conjuntos de *peers* que trabalham cooperativamente para receber os pedaços são denominados de *swarms* [Cohen 2003].

O exposto é a motivação para este artigo, cujo objetivo é a proposta de uma nova adaptação do BitTorrent para *streaming* de vídeo sob demanda ante clientes interativos em MANETs. Esta proposta é denominada BitTorrent Interativo com *Broadcast* - Rede (BIB-R), e possui, como aspectos mais inovadores, o emprego de tráfego *broadcast* e a avaliação das rotas de disseminação do vídeo conforme o número de saltos constituintes. Por meio de simulações, onde são mensuradas diferentes métricas e realizadas comparações com outras propostas da literatura, a adaptação BIB-R é validada. Nesse sentido, a principal contribuição deste artigo é prover novos subsídios para o desenvolvimento de protocolos que possam oferecer adequados níveis de QoS (*Quality of Service*) do sistema e de QoE (*Quality of Experience*) do cliente, considerando *streaming* de vídeo sob demanda interativo em MANETs.

O restante deste artigo é organizado como segue. A Seção 2 revisa a operação do BitTorrent, além de explicar suas limitações de emprego com respeito a redes sem fio e à realização de *streaming*. Na Seção 3, tem-se uma revisão da literatura. A Seção 4 discorre sobre os critérios utilizados para as seleções de *peers* e de pedaços. A proposta BIB-R é apresentada na Seção 5. A Seção 6 apresenta experimentos, resultados e análises. Finalmente, conclusões gerais e trabalhos futuros constituem a Seção 7.

2. Fundamentos

2.1. Operação Básica do BitTorrent

Seja p um *peer* que deseja receber um arquivo F , remotamente armazenado em um servidor da Internet. O *peer* p deve juntar-se a um *swarm* S que esteja replicando o arquivo F . Para tanto, o *peer* p contata o *tracker*, que é o coordenador da comunicação entre os *peers* do *swarm* S . Após contatado, o *tracker* envia ao *peer* p uma lista L , que contém *peers* pertencentes ao *swarm* S . O *peer* p tenta então aleatoriamente estabelecer conexões TCP com os *peers* da lista L . Os *peers* com os quais ocorrem conexões bem-sucedidas formam o conjunto de *vizinhos* do *peer* p . Os *vizinhos* são então os *peers* para os quais, e a partir dos quais, o *peer* p pode enviar, e receber, pedaços do arquivo F .

Seja g um *peer* que é *vizinho* do *peer* p . Diz-se que o *peer* g está interessado no *peer* p quando o *peer* p tem pedaços do arquivo F que o *peer* g não possui. Opostamente, diz-se que o *peer* g não está interessado no *peer* p quando o *peer* p possui apenas um subconjunto dos pedaços do *peer* g . Como mencionado, a política de seleção de *peers*

e a política de seleção de pedaços orientam o processo de transmissão dos pedaços do arquivo F pela rede, sendo estas explicadas a seguir.

A política de seleção de *peers*, também chamada de algoritmo de *choke*, é utilizada para que o *peer p* decida que outros *peers*, dentre aqueles vizinhos que estão nele interessados, serão autorizados (i.e., *unchoked*) a requisitar seus pedaços. Essa política é constituída pelos seguintes dois processos independentes [Cohen 2003].

1. A cada intervalo de $\delta t = 10$ s: os três *peers* que em média fornecem pedaços para o *peer p* com as mais altas taxas de *upload* são os escolhidos. Este processo é chamado de *regular unchoking*;
2. A cada intervalo de $\delta t = 30$ s: um quarto *peer* é escolhido aleatoriamente. Este processo é chamado de *optimistic unchoking*.

A política de seleção de pedaços, por sua vez, é usada para que um *peer g* decida que pedaços solicitar ao ser autorizado (i.e., *unchoked*) pelo seu *peer* vizinho p . Para tanto, o *peer g* mantém uma lista do número de cópias de cada pedaço no seu conjunto de vizinhos. Essa informação é usada para definir um conjunto de pedaços mais raros (i.e., menos replicados). Seja m o número de cópias do pedaço mais raro entre os vizinhos. O índice de cada pedaço com m cópias é então adicionado ao conjunto de pedaços mais raros. Após ser autorizado, o *peer g* solicita o próximo pedaço considerando seu conjunto de pedaços mais raros e os pedaços disponíveis localmente no *peer p*. Após receber o pedaço solicitado, o *peer g* avisa a todos os seus vizinhos sobre o fato ocorrido [Cohen 2003].

2.2. Limitações do BitTorrent

O protocolo BitTorrent não considera a topologia física da rede subjacente. Suas políticas de seleção de pedaços e de seleção de *peers* atuam exclusivamente sobre uma topologia lógica, provida pela camada de aplicação. A razão para tanto é que sua concepção é pensada para redes com fio, caracterizadas por canais de comunicação estáveis e baixa perda de pacotes de dados [Rethfeldt et al. 2018].

Ao se usar o BitTorrent em uma rede sem fio, a seleção de *peers* considerando a topologia lógica pode deteriorar o desempenho do sistema. Isso porque, sob a topologia lógica, os *peers* vizinhos aparecem como se estivessem diretamente conectados, i.e., enlace direto de um único salto, mas eles podem estar a múltiplos saltos entre si na real topologia física, i.e., enlace de múltiplos saltos, usando *peers* intermediários [Rethfeldt et al. 2018].

Neste contexto, há dois paradigmas de desenvolvimento possíveis de adotar para implementar adaptações do BitTorrent: *layered architecture* e *cross layer*. No primeiro, a operação da rede é dividida em camadas distintas e define-se uma hierarquia de serviços a serem providos pelas camadas individualmente. Inexiste a comunicação direta entre camadas não adjacentes, e a comunicação entre camadas adjacentes restringe-se a chamadas de procedimentos e respostas. Simplicidade é a principal vantagem, mas não se adquire a consciência da topologia física da rede subjacente. No segundo, a hierarquia do modelo em camadas é flexibilizada, permitindo que camadas não adjacentes se comuniquem para obter melhores desempenhos. Neste caso, as implementações são mais complexas, podendo estabelecer dependência de informações entre várias camadas, mas consegue-se a consciência da topologia física da rede subjacente [Mantzouratos et al. 2012].

Uma outra limitação importante é que, sob BitTorrent, a replicação de pedaços não é feita de forma estritamente sequencial (i.e., gulosa), pois a política de seleção de pedaços prioriza os pedaços mais raros (i.e., menos replicados). Isso impede, portanto, a reprodução contínua do arquivo desde o início de sua recepção, inviabilizando aplicações caracterizadas por *streaming*. Para mitigar essa limitação, as adaptações do BitTorrent usualmente fazem uso de uma *janela deslizante*, a qual delimita os pedaços possíveis de serem solicitados pelo *peer* [Rodrigues 2018].

3. Revisão da Literatura

As adaptações do BitTorrent para redes sem fio majoritariamente consideram apenas o serviço de replicação de arquivos, sem interatividade do cliente e, muitas vezes, sem considerar a mobilidade dos nós. A discussão concentra-se mais na otimização do tempo de *download* do arquivo, sem qualquer preocupação com a reprodução no lado do cliente. As propostas mesclam modificações nas políticas de seleção de *peers* e de seleção de pedaços. Como exemplos, citam-se os seguintes dois trabalhos: [Sbai and Barakat 2009, Rethfeldt et al. 2018]. Neste contexto, uma exceção é o trabalho de [Rodrigues and Rocha 2019a], que trata sobre *streaming* de vídeo sob demanda, mas não considera a interatividade do cliente.

Tratando-se especificamente de *streaming* sob demanda interativo em MANETs, o número de obras é bem limitado, existindo apenas, salvo melhor juízo, os quatro trabalhos discutidos a seguir, sendo apenas um deles não baseado no BitTorrent. Primeiro, em [Rodrigues 2018], tem-se o Algoritmo de Distância Mínima com Reciprocidade Indireta (AD-RI). De paradigma *layered architecture* e baseado no BitTorrent, a seleção de *peers* considera *taxa de upload*, *distância geográfica*, e *reciprocidade indireta*. A seleção de pedaços, por sua vez, considera o uso de uma *janela deslizante*, delimitando dinamicamente os pedaços possíveis de serem solicitados. Os resultados obtidos por meio de simulação demonstram a eficiência da proposta comparativamente a esquemas ideais de transmissão. Segundo, em [Rodrigues and Rocha 2019c], é apresentado o algoritmo BitTorrent Interativo para MANET (BTM-I). De paradigma *layered architecture* e baseado no BitTorrent, BTM-I diferencia-se de AD-RI por implementar a transmissão de pedaços via *unicast* e *broadcast* e, ainda, por não considerar a *reciprocidade indireta* como critério. As simulações realizadas mostram uma maior eficiência de BTM-I com relação à AD-RI.

Terceiro, em [Rodrigues and Rocha 2019b], são propostos dois algoritmos baseados no BitTorrent: BitTorrent Interativo de Aplicação (BTI-A) e BitTorrent Interativo de Rede (BTI-R). Ambos utilizam apenas transmissão *unicast*, sendo que BTI-A é de paradigma *layered architecture*, enquanto BTI-R é de paradigma *cross layer*. Diferenciando-se de BTM-I e AD-RI, a seleção de *peers* de BTI-A e BTI-R é de concepção mais simples, pois utiliza apenas taxas de *upload* e *download* como critérios de decisão. Como importante constatação, as análises realizadas evidenciam o fato de que *peers* alteram dinamicamente seus desempenhos devido às instabilidades das conexões sem fio. As simulações então conduzidas destacam um atrativo desempenho de BTI-A e BTI-R, sendo ambos superiores à AD-RI e, em certos cenários, também ao *Delay-Sensitive Segment Scheduling Algorithm* (DSSSA) [Hu et al. 2017], que é o quarto e último trabalho, explicado a seguir.

A proposta DSSSA segue o paradigma *cross layer*, sendo a única não baseada no BitTorrent. Sua operação consiste em escalonar os pedaços do arquivo na seguinte

sequência. Primeiro, são escalonados os pedaços com instantes de visualização bem próximos do instante corrente (i.e., pedaços urgentes). Segundo, são escalonados os pedaços que não são urgentes, considerando os correspondentes instantes de visualização (i.e., pedaços normais). Terceiro, são escalonados os pedaços que estão replicados apenas em alguns poucos *peers* (i.e., pedaços mais raros). Além disso, como um mesmo pedaço pode estar armazenado em distintos *peers*, seleciona-se a rota mais rápida (i.e., de menor congestionamento). Apesar da eficiência observada, as simulações abrangem cenários limitados a 50 *peers*, sem mencionar que algum *overhead* de processamento deve existir devido ao cálculo *online* da prioridade dos pedaços e da determinação das rotas mais rápidas. Além disso, a mobilidade e a interatividade são examinadas de maneira modesta.

Ante o exposto, este trabalho se reveste de relevância e ineditismo pelos dois pontos a seguir. Primeiro, como mencionado, há apenas quatro trabalhos anteriores que tratam sobre *streaming* de vídeo sob demanda interativo em MANETs. Ademais, todos datam a partir do ano de 2017. Isso evidencia a atualidade deste problema de pesquisa, bem como a importância de trabalhos subsequentes para ratificação dos resultados alcançados e ampliação dos escopos considerados. Segundo, a nova proposta BIB-R traz a implementação conjunta dos mais promissores critérios de seleção (de *peers* e de *pedaços*) utilizados nos quatro trabalhos anteriores, com destaque para o emprego de tráfego *broadcast* e a seleção de rotas em função do número de saltos existente. Além disso, a validação de BIB-R é feita considerando análises comparativas com os quatro trabalhos anteriores, garantindo um amplo e completo cenário de competição.

4. Critérios de Seleção

Esta seção discorre sobre os critérios mais promissores para as seleções de *peers* e de pedaços nas adaptações do BitTorrent para *streaming* em MANETs. Para maior facilidade de entendimento, a exposição é feita por meio de duas subseções, onde admite-se que g é um *peer* vizinho de um outro *peer* p em um *swarm* e, ainda, que o *peer* g está correntemente interessado no *peer* p .

4.1. Seleção de Peers

Os seguintes critérios são admitidos pelo *peer* p para seleção do *peer* g :

Taxa de upload: dentre os vizinhos interessados no *peer* p , o *peer* g deve ser aquele que mais rapidamente envia pedaços para o *peer* p em média. Em caso de empate, a seleção é aleatória. Este critério reforça a reciprocidade direta, pois apenas aqueles *peers* que contribuem diretamente para o *peer* p podem receber pedaços a partir dele. Daí, *peers* egoístas, i.e., *peers* que apenas querem receber pedaços, são eliminados do sistema. O efeito colateral é que *peers* lentos, i.e., *peers* de baixa capacidade de transmissão, terão dificuldades em participar da replicação, passando a depender da seleção proveniente do processo *optimistic unchoking*.

Reciprocidade indireta: dentre os *peers* interessados no *peer* p , o *peer* g deve ser aquele que mais envia pedaços em média a outros *peers* distintos do *peer* p , i.e., não se consideram os pedaços enviados diretamente para o *peer* p . Em caso de empate, a seleção é aleatória. Este critério busca uniformizar a distribuição de dados na rede, evitando que apenas *peers* de alta capacidade de transmissão possam receber pedaços. De certa forma, este critério ameniza o efeito colateral do critério anterior (i.e., *taxa de upload*).

Contagem de saltos: dentre os vizinhos interessados no *peer p*, o *peer g* deve ser o mais próximo do *peer p*, em termos do número de saltos constituintes da rota física até o *peer p*. Em caso de empate, a seleção é aleatória. A motivação para este critério é a eliminação do *overhead* de processamento para estabelecer rotas longas em número de saltos. Estas rotas frequentemente terminam sendo inúteis, pois tendem a ser mais lentas que rotas mais curtas para a transmissão de pedaços entre um mesmo par de *peers* que sejam os extremos (i.e., fonte e destino) da rota. Além disso, este critério termina por otimizar o uso da capacidade de transmissão dos nós intermediários da rota.

4.2. Seleção de Pedacos

O critério mais utilizado para a seleção de pedaços resulta da combinação de uma janela deslizante W com um *buffer* interior V , conforme explicado a seguir.

A janela W delimita dinamicamente o conjunto de pedaços que podem ser solicitados pelo *peer g* ao *peer p*. Por sua vez, o *buffer* interior V delimita um subconjunto de pedaços dentro da janela W , estabelecendo uma prioridade dentre os pedaços contidos em W . No caso de todos os pedaços de V já terem sido recuperados pelo *peer g*, então o próximo pedaço a ser solicitado é o pedaço mais raro contido em W que ainda não foi recuperado; caso contrário, o próximo pedaço a ser solicitado pelo *peer g* é o próximo pedaço consecutivo dentro de V que ainda não foi recuperado. A janela W e o *buffer* V sempre iniciam no pedaço do vídeo que está em reprodução pelo cliente, sendo atualizados dinamicamente.

Este critério estabelece um compromisso entre a diversidade de pedaços no *swarm*, responsável pela exemplar eficiência do BitTorrent original, e a continuidade na reprodução do vídeo no lado do cliente, mandatória para atingir-se uma satisfatória QoE.

5. Nova Adaptação BIB-R

Esta seção explica a nova adaptação BIB-R por meio de suas políticas de seleção de *peers* e de seleção de pedaços.

5.1. Política de Seleção de *Peers*

Admita a visão de um *peer p* que acaba de se conectar ao sistema. A capacidade de *upload* do *peer p* é dividida em $(y + x)$ *slots* de dados: y *slots* para o processo *limited regular unchoking*, e x *slots* para o processo *limited altruistic unchoking*. Para *limited regular unchoking*, que ocorre a cada δt_R segundos, tem-se os seguintes passos.

1. Os vizinhos interessados no *peer p*, que estão dentro do limite de h saltos de distância, são ordenados de acordo com a *taxa de upload* média;
2. Os $(y - 1)$ vizinhos de maiores taxas são então selecionados pelo *peer p*, e um *slot* de dados é alocado para cada um deles. A prioridade é, portanto, para aqueles que mais rapidamente fornecem pedaços para o *peer p*;
3. Os k primeiros vizinhos interessados remanescentes são então ordenados em função da *reciprocidade indireta* média, onde $k < (y - 1)$;
4. O vizinho interessado de maior valor de *reciprocidade indireta* média é então selecionado pelo *peer p*, e um *slot* de dados é alocado para ele.

Note que os vizinhos são selecionados pelo *peer p* considerando os três critérios discutidos anteriormente: *taxa de upload*, *reciprocidade indireta* e *contagem de saltos*. O primeiro valor é computado pelo próprio *peer p*. O segundo valor é informado ao *peer p* pelo correspondente vizinho interessado. O terceiro valor é obtido pelo *peer p* a partir do protocolo de roteamento da rede, caracterizando o uso da abordagem *cross layer*.

A seguir tem-se os passos para o processo *limited altruistic unchoking*, que ocorre a cada δt_A segundos. Perceba que, neste caso, a prioridade é para os vizinhos interessados que mais rapidamente conseguem receber pedaços a partir do *peer p*.

1. Os vizinhos interessados no *peer p*, que estão a mais de h saltos de distância, são ordenados de acordo com a *taxa de download* média;
2. Os x vizinhos interessados de maiores taxas são então selecionados pelo *peer p*, e um *slot* de dados é alocado para cada um deles.

5.2. Política de Seleção de Pedaços

A explicação que segue considera que: os *peers p* e *g* pertencem ao mesmo *swarm* em uma MANET, o *peer g* é um vizinho do *peer p*, e o *peer g* está interessado no *peer p*. O critério utilizado nesta política é aquele que combina a janela deslizante, W , e o *buffer* interior, V , discutido anteriormente.

Admita que o vídeo é dividido em l pedaços. Seja d o pedaço correspondente ao ponto atual de reprodução. Seja w o tamanho da janela deslizante W , em número de pedaços. Daí, W abrange os seguintes pedaços: $[d; d + w]$. Além disso, W é dinamicamente atualizada conforme andamento da reprodução do vídeo. Por exemplo, seja Δ_{play} o número de pedaços consecutivos que acabam de ser visualizados. O primeiro pedaço de W é então atualizado para $(d + \Delta_{play})$, e o último pedaço é atualizado para $(d + w + \Delta_{play})$.

Seja v o tamanho do *buffer* interno V , em número de pedaços. O primeiro pedaço de V é sempre coincidente com o primeiro pedaço de W . Além disso, antes de o *peer g* solicitar o próximo pedaço ao *peer p*, é sempre verificado se todos os pedaços de V foram recuperados. Em caso afirmativo, o *peer g* solicita o pedaço mais raro pertencente a W que ainda não foi recuperado; caso contrário, o *peer g* solicita o próximo pedaço consecutivo dentro de V ainda não recuperado.

Os pedaços transmitidos por meio de *peers* intermediários são localmente armazenado nesses *peers*. Isso é feito para compensá-los pelo uso de suas capacidades de transmissão para enviar pedaços ao longo de rotas que conectam *peers* extremos (i.e., origem e destino). Por fim, sempre que um *peer* do *swarm* transmite um pedaço pela primeira vez, este pedaço é necessariamente transmitido via *broadcast*.

6. Avaliação de Desempenho

Os experimentos são realizados via simulação e considerando uma MANET composta por n *peers*. Cada *peer* atua como um cliente móvel que realiza ações de interatividade durante a reprodução do arquivo de vídeo. A capacidade de transferência de dados individual (*upload* e *download*) de cada *peer* é 4 Mbps. Entre os n *peers*, há um *tracker* e, pelo menos, um *seed* que operam de forma contínua e são diretamente alcançáveis pelos demais *peers* do sistema. O restante desta seção é dividido nas três subseções a seguir.

Tabela 1. Perfis de interatividade.

Parâmetro	Baixa (IB)	Alta (IA)
λ	0,005/s	0,025/s
l_{play} l_{jump} l_{pause}	14,5% de f_{size}	1,5% de f_{size}
$p_0; p_1$	0,89; 0,01	0,55; 0,15
$p_2; p_3$	0,05; 0,05	0,15; 0,15

6.1. Mobilidade e Interatividade

Em se tratando da mobilidade, os *peers* se movimentam em uma área plana de 100 m², livre de obstáculos, com velocidade entre 0 e 2 m/s, e com direção entre 0 e 360°. Essa mobilidade é definida a partir do modelo de mobilidade *Small World in Motion* (SWIM) [Mei and Stefa 2009]. Este modelo baseia-se nas seguintes três premissas: (i) o indivíduo tende a visitar lugares populares ou próximos do lugar onde passa a maior parte do tempo; (ii) o indivíduo permanece a maior parte do tempo em poucos lugares (e.g., lares, escritórios e escolas), enquanto permanece por um curto período de tempo em vários lugares (e.g., bancos, restaurantes e *shopping centers*); (iii) finalmente, a velocidade do indivíduo depende de quão próximo ele está do destino final. Os parâmetros utilizados para configurar o simulador são os mesmos considerados em [Mei and Stefa 2009].

Para a interatividade, emprega-se o modelo considerado em [Rodrigues 2018]. As ações de interatividade podem ser: *Play*, *Pause*, *Jump Forwards* (JF), e *Jump Backwards* (JB). A ação *Play* representa a reprodução do vídeo pelo cliente. A ação *Pause* indica que a reprodução é pausada. As ações JB e JF representam um salto a um ponto de reprodução anterior e posterior ao atual, respectivamente. Ademais, as ações *Play*, *Pause*, JF e JB são geradas de acordo com uma distribuição de *Poisson* com taxa λ e com probabilidades p_0 , p_1 , p_2 e p_3 , respectivamente. As ações *Play* e *Pause* possuem tamanhos l_{play} e l_{pause} , respectivamente. As ações JF e JB possuem o mesmo tamanho l_{jump} . Esses tamanhos são um percentual do tamanho total do arquivo de vídeo f_{size} . Por fim, os dois perfis de interatividade considerados nas simulações são *interatividade baixa* (IB) e *interatividade alta* (IA). A Tabela 1 lista os valores numéricos dos parâmetros de configuração mencionados, que são os mesmos utilizados em [Rodrigues 2018, Rodrigues and Rocha 2019c].

6.2. Simulação e Métricas

Os experimentos são realizados no simulador PeerSim [Montresor and Jelasity 2009]. A plataforma computacional é um Intel Core i7 (2.6 GHz), 24 GB de RAM, executando o sistema operacional GNU/Linux. Os resultados da simulação têm intervalos de confiança de 95% que estão dentro do limite de 5% dos valores médios reportados, tendo sido realizadas 10 execuções (rodadas).

Na camada de rede, o roteamento de pacotes é baseado no protocolo *Ad hoc On-Demand Distance Vector* (AODV) do simulador WiFi Direct [Baresi et al. 2016]. Para as camadas MAC/PHY, são considerados os protocolos CSMA/CA e 802.11, respectivamente, com um percentual de perda de pacotes de 1,0%. O arquivo a ser distribuído por *streaming* é um vídeo codificado a 300 kbps. Este arquivo é dividido em 390 pedaços de

256 kB, sendo que cada pedaço é dividido em blocos de 16 kB. Assim, o vídeo possui um tamanho aproximado total de 100 MB. Apesar de o bloco ser a unidade fundamental de envio pela rede física, a análise considera somente os pedaços transmitidos.

Para fins de análise e validação da nova adaptação BIB-R, são consideradas as seguintes duas métricas de desempenho: *download operacional* (DO), que estima a taxa média do *peer* para receber o vídeo; e o *tempo de descontinuidade* (TD), que estima o tempo médio total de interrupção do *peer* durante a reprodução do vídeo. A análise conjunta dessas duas métricas permite inferir tanto sobre a QoS entregue pelo sistema, quanto sobre a QoE percebida pelo cliente.

6.3. Resultados e Análises

A seguir analisam-se as propostas BIB-R, BTM-I [Rodrigues and Rocha 2019c], BTI-A, BTI-R [Rodrigues and Rocha 2019b] e DSSSA [Hu et al. 2017]. Consideram-se o modelo SWIM, as duas métricas de desempenho e os dois perfis de interatividade, explicados anteriormente. Além disso, assumem-se os seguintes valores de configuração: $v = 30$, $w = 390$, $\delta t_R = 200$, $\delta t_A = 200$, $y = 3$, $x = 1$, $k = 2$, e $h = 2$. Estes valores são idealmente determinados experimentalmente e inspirados pelos trabalhos de [Sbai and Barakat 2009, Rodrigues and Rocha 2019c, Rodrigues and Rocha 2019b].

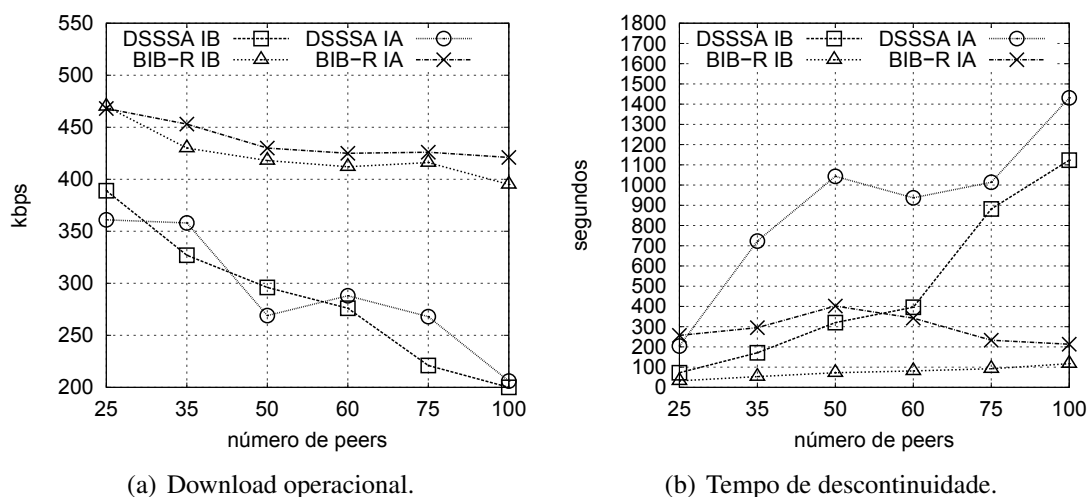


Figura 1. Comparando BIB-R e DSSSA no modelo de mobilidade SWIM.

A Figura 1 compara BIB-R e DSSSA em função do número de *peers* na rede. Para $n \leq 100$, BIB-R possui um desempenho superior na métrica DO , independentemente do perfil de interatividade. Com respeito à métrica TD , tem-se que BIB-R segue sendo superior, excetuando-se para $n = 25$ no perfil IA. Em complemento, a Tabela 2 (colunas 1 e 2) mostra que BIB-R proporciona um ganho em DO de até 51%, e uma redução em TD de até 89%. O valor negativo nesta tabela refere-se ao caso de redes de tamanho pequeno (i.e., $n = 25$), onde DSSSA supera BIB-R na métrica TD no perfil IA. Neste caso específico, como as rotas nas redes pequenas apresentam menor atraso (pelo pouco congestionamento), o critério da escolha da rota mais rápida, usado pelo DSSSA, resulta mais efetivo que o critério da distância, em número de saltos, usado pelo BIB-R. De forma geral, a razão do melhor desempenho de BIB-R nessa comparação reside na utilização do paradigma descentralizado para decidir a quais vizinhos solicitar os pedaços.

A Figura 2 compara BIB-R e BTM-I conforme o número de *peers* na rede. BIB-R possui um desempenho superior nas métricas *DO* e *TD*, independentemente do perfil de interatividade, exceto para $n = 50$ na métrica *TD* no perfil IA. Em complemento, a Tabela 2 (colunas 3 e 4) mostra que BIB-R provê um ganho em *DO* de até 15%, e uma redução em *TD* de até 64%. O valor negativo nesta tabela refere-se ao caso específico onde BTM-I supera ligeiramente BIB-R em *TD*. Em geral, o melhor desempenho de BIB-R nessa comparação se dá pela utilização da *reciprocidade indireta*, pois *peers* que não conseguem compartilhar muitos pedaços (e.g., devido à instabilidade das conexões sem fio) terminam não sendo tão penalizados quanto sob BTM-I.

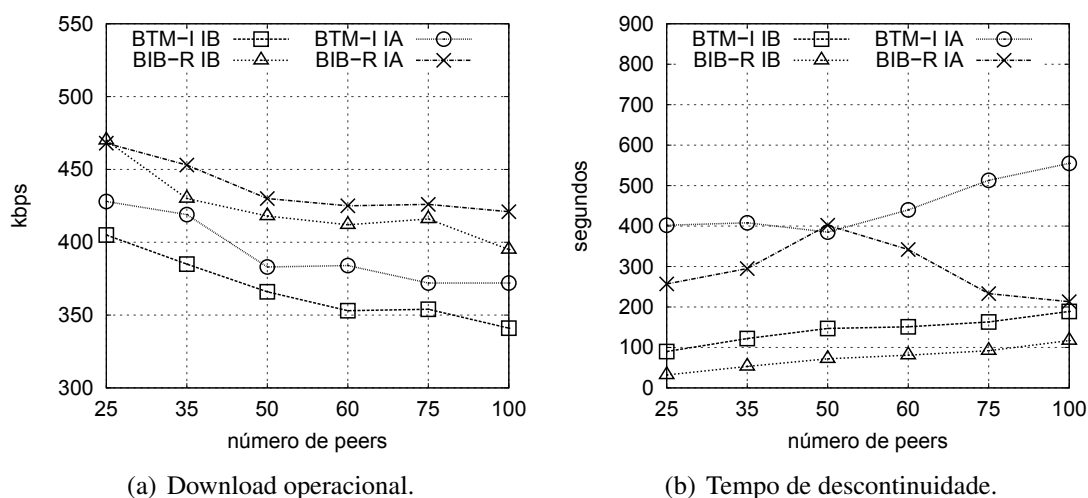


Figura 2. Comparando BIB-R e BTM-I no modelo de mobilidade SWIM.

Por restrição de espaço, aqui não são apresentadas figuras para os resultados da comparação envolvendo BIB-R, BTI-A e BTI-R. Para a métrica *DO*, BIB-R tem sempre o melhor desempenho, independentemente do perfil de interatividade. Para a métrica *TD*, BIB-R é superada somente no caso específico da comparação com BTI-R em redes de tamanhos médio e grande (i.e., $n > 25$) no perfil IB. Em complemento, na Tabela 2 (colunas 5 à 8), mostra-se que BIB-R proporciona ganhos em *DO* de até 14% e 18%, e reduções de até 53% e 69% em *TD*, em relação à BTI-R e BTI-A, respectivamente. O valor negativo em *TD* refere-se ao caso mencionado em que BIB-R é superada por BTI-R. Em geral, o desempenho superior de BIB-R nessa comparação ocorre pelo uso do tráfego *broadcast*, que permite a disseminação mais efetiva dos pedaços.

Tabela 2. Otimizações alcançadas por BIB-R sob o modelo SWIM.

para DSSSA		para BTM-I		para BTI-R		para BTI-A		–
Min	Max	Min	Max	Min	Max	Min	Max	Métrica
17%	51%	7%	15%	3%	14%	13%	18%	<i>DO</i>
-20%	89%	-4%	64%	-56%	53%	30%	69%	<i>TD</i>

Por fim, a Tabela 3 traz uma síntese da classificação geral do desempenho das propostas analisadas. BIB-R resulta como a mais competitiva, ficando em primeiro ou, no pior caso, em segundo lugar. Essa classificação é condizente com os resultados apresentados nos trabalhos relacionados, haja vista que BIB-R implementa uma combinação

inteligente dos mais promissores critérios em prol da otimização de desempenho. Especificamente, na métrica *DO*, a superioridade de BIB-R é absoluta. Além disso, o valor observado dessa métrica está sempre acima da taxa de codificação do vídeo (i.e., 300 kbps), assegurando uma satisfatória QoS do sistema. Com respeito à *TD*, os resultados mostram que BIB-R, mesmo quando em desvantagem em relação a alguma das outras propostas, apresenta descontinuidades correspondentes em média a apenas 2,8% e 10,9% do tempo total de reprodução nos perfis IB e IA, respectivamente, redundando em uma aceitável QoE do cliente. Não obstante, se necessário, pode-se buscar mitigar essas descontinuidades por meio de, e.g., técnicas de *buffering*, *prefetching* e/ou intercalação.

Tabela 3. Análise competitiva sob o modelo SWIM.

Perfil	IB			IA				
	<i>DO</i>	<i>TD</i>		<i>DO</i>		<i>TD</i>		
Tamanho	$n \leq 100$	$n = 25$	$n > 25$	$25 < n \leq 50$	<i>caso contrário</i>	$n = 25$	$25 < n < 75$	$n \geq 75$
1º	BIB-R	BIB-R	BTI-R	BIB-R	BIB-R	DSSSA	BIB-R	BIB-R
2º	BTI-R	BTI-R	BIB-R	BTI-R	BTM-I	BIB-R	BTM-I	BTI-R
3º	BTM-I	DSSSA	BTM-I	BTM-I	BTI-R	BTM-I	BTI-R	BTM-I
4º	BTI-A	BTM-I	BTI-A	BTI-A	BTI-A	BTI-R	BTI-A	BTI-A
5º	DSSSA	BTI-A	DSSSA	DSSSA	DSSSA	BTI-A	DSSSA	DSSSA

7. Conclusões e Trabalhos Futuros

Este artigo propôs uma adaptação do BitTorrent, denominada de BIB-R, visando o serviço de *streaming* de vídeo sob demanda em MANETs ante clientes interativos. Usando simulações, constatou-se a eficiência da proposta e concluiu-se positivamente sobre a QoS do sistema e a QoE do cliente. Como resultados pontuais, destacam-se: a recuperação de dados e a continuidade na reprodução ficaram em patamares atrativos; a escalabilidade foi confirmada para redes de até 100 *peers*; por fim, o envio de dados em *broadcast* e a seleção das rotas em função do número de saltos se revelaram promissores para emprego em novos protocolos de *streaming*. Como trabalhos futuros e contemplando limitações deste artigo, apontam-se: complementar a análise com modelos analíticos e medições reais; avaliar a proposta para *live streaming* e com uso de *distributed hash table* (DHT); e realizar comparações com outras propostas da literatura ante, e.g., diferentes modelos de mobilidade e interatividade, áreas de cobertura, roteamentos, e codificações de arquivo (e.g., [Correa da Silva et al. 2019, Carvalho et al. 2018, Assis et al. 2019]).

Referências

- Assis, F., Nogueira, M., Menasché, D., Pinheiro, J., Sermpezis, P., Kastanakis, S., Spyropoulos, T., and Delgado, C. (2019). Recomendação de Conteúdo e QoE: Um Experimento Quantificando o Papel da QoS nas Preferências por Vídeos. In *18º WPerformance/CSBC 2019*, Belém, PA, Brasil.
- Baresi, L., Derakhshan, N., and Guinea, S. (2016). WiDiSi: A Wi-Fi direct simulator. In *IEEE Wireless Communications and Networking Conference*, Doha, Qatar.
- Carvalho, D. et al. (2018). Understanding users-content interaction in non-linear multimedia streaming services. In *Brazilian Symposium on Multimedia and the Web (Web-Media'18)*, Salvador, Bahia, Brazil.

- CISCO (2019). Cisco visual networking index: Forecast and trends, 2017–2022. *Cisco White Paper*. Document ID:1551296909190103.
- Cohen, B. (2003). Incentives build robustness in BitTorrent. In *First Workshop on Economics of Peer-to-Peer System*, Berkeley, USA.
- Correa da Silva, D. V., Velloso, P. B., and Rocha, A. A. (2019). Using data mining techniques to extract key factors in mobile live streaming. In *2019 IEEE Symposium on Computers and Communications (ISCC)*, pages 1–6.
- Fleury, M., K. D. and Qadri, N. N. (2019). Video streaming over MANETs: An overview of techniques. *Multimedia Tools Applications*, 6:23749–23782.
- Hu, C.-C., Lai, C.-F., Hou, J.-G., and Huang., Y.-M. (2017). Timely scheduling algorithm for P2P streaming over MANETs. *Computer Networks*, 127:56 – 67.
- Machado, B., Vieira, A., Cunha, I., and Ziviani, A. (2019). Evolução do Comportamento do Usuário em Eventos de Larga Escala na Internet. In *18º Wperformance/CSBC 2019*, Belém, PA, Brasil.
- Mantzouratos, S. et al. (2012). Survey of cross-layer proposals for video streaming over Mobile Ad hoc Networks (MANETs). In *2012 International Conference on Telecommunications and Multimedia (TEMU)*, pages 101–106, Chania, Greece.
- Mei, A. and Stefa, J. (2009). SWIM: A Simple Model to Generate Small Mobile Worlds. In *IEEE INFOCOM*, Rio de Janeiro, RJ, Brazil.
- Montresor, A. and Jelasity, M. (2009). PeerSim: A scalable P2P simulator. In *IEEE Ninth International Conference on Peer-to-Peer Computing*, Washington, USA.
- Rethfeldt, M. et al. (2018). MeNTor: A wireless-mesh-network-aware data dissemination overlay based on bittorrent. *Ad Hoc Networks*, 79:146 – 159.
- Rodrigues, C. K. S. (2018). Efficient BitTorrent-Like Algorithms for Interactive On-Demand Multimedia Streaming over MANETs. In *Brazilian Symposium on Multimedia and the Web (WebMedia'18)*, Salvador, Bahia, Brazil.
- Rodrigues, C. K. S. and Rocha, V. (2019a). BT-MANET: A Novel BitTorrent-Like Algorithm for Video On-Demand Streaming over MANETs. *IEEE Latin America Transactions*, 17(01):78–84.
- Rodrigues, C. K. S. and Rocha, V. (2019b). Towards adapting BitTorrent for interactive on-demand multimedia streaming over MANETs. In *Brazilian Symposium on Multimedia and the Web (WebMedia'19)*, Rio de Janeiro, RJ, Brazil.
- Rodrigues, C. K. S. and Rocha, V. (2019c). Uma Adaptação do BitTorrent para Streaming de Vídeo sob Demanda Interativo em Redes Móveis Ad Hoc. In *18º WPerformance/CSBC 2019*, Belém, PA, Brasil.
- Sbai, M. K. and Barakat, C. (2009). Revisiting P2P Content Sharing in Wireless Ad Hoc Networks. In *IWSOS*, Zurich, Switzerland.
- Wang, J., Xie, B., and Agrawal, D. P. (2009). *Journey from Mobile Ad Hoc Networks to Wireless Mesh Networks*, pages 1–30. Springer London, London.