

Uma Adaptação do BitTorrent para *Streaming* de Vídeo sob Demanda Interativo em Redes Móveis *Ad Hoc*

Carlo Kleber da Silva Rodrigues¹, Vladimir Emiliano Moreira 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 paper proposes an adaptation of the peer-to-peer BitTorrent algorithm to perform interactive on-demand video streaming over mobile ad hoc networks. This adaptation mainly lies on the idea of prioritizing the sharing of data between peers which are geographically close to each other, as well as restricting the data that may be requested by peers during video playback. Through simulations and assessing different metrics, the proposed adaptation is validated and its attractive performance is proven. In this sense, the main contribution of this article is to provide a new algorithmic solution for the design of real protocols targeted at mobile ad hoc networks. At last, overall conclusions and future work conclude this article.*

Resumo. *Este artigo propõe uma adaptação do algoritmo peer-to-peer BitTorrent para realizar streaming de vídeo sob demanda interativo em redes móveis ad hoc. Em linhas gerais, essa adaptação baseia-se na ideia de priorizar o compartilhamento de dados entre peers geograficamente próximos entre si, bem como restringir os dados que podem ser requisitados pelos peers durante a reprodução do vídeo. Por meio de simulações e ante diferentes métricas, a adaptação proposta é validada e comprova-se a sua atrativa performance. Nesse sentido, a principal contribuição deste artigo é prover uma nova solução algorítmica para a concepção de protocolos para redes móveis ad hoc. Conclusões gerais e trabalhos futuros encerram este artigo.*

1. Introdução

O algoritmo *peer-to-peer* (P2P) BitTorrent é reconhecido pela sua eficiência para a replicação de arquivos na Internet [Cohen 2003; Legout; Urvoy-Keller; e Michiardi 2006]. A filosofia deste algoritmo consiste em dividir um arquivo em pedaços e fazer com que os *peers* de um certo grupo, que desejem receber este arquivo, passem então a trabalhar cooperativamente, transmitindo os pedaços entre si.

Mais especificamente, duas políticas formam o núcleo do BitTorrent: a seleção de *peers* e a seleção de pedaços. A primeira determina quais os *peers* que devem ser autorizados a receber os pedaços que um outro *peer* possui, e a segunda seleciona os pedaços que devem ser requisitados por um *peer*. No estado de equilíbrio, o processo de transmissão de pedaços obedece ao seguinte: *quem mais compartilha os pedaços já recebidos, maior probabilidade tem de receber outros novos pedaços*. Esse é o paradigma da operação geral do BitTorrent [Legout; Urvoy-Keller; e Michiardi 2006].

A eficiência do BitTorrent logo motivou propostas de sua adaptação para o serviço de *streaming* sob demanda em redes com fio, também chamadas de redes cabeadas (e.g., [D’Acunto et al. 2013]). A adaptação se faz mandatória porque, sob o BitTorrent, os pedaços não são requisitados e recebidos sequencialmente, o que impede a reprodução contínua desde o início da recepção.

Considerando que os dispositivos sem fio (e.g., *smartphones* e *notebooks*) estão sendo cada vez mais utilizados e que *streaming* tende a seguir como a principal tecnologia para distribuição de conteúdo multimídia [Domingues et al. 2018], a natural evolução do BitTorrent é então sua adaptação para *streaming* em redes sem fio, especialmente com interatividade dos clientes. Corroborando esse pensamento, há previsões de que *streaming* de vídeo sob demanda e *download* de arquivos juntos corresponderão a mais de 81% do tráfego IP da Internet em 2021 e, nesse mesmo ano, o tráfego de dispositivos sem fio equivalerá a mais de 63% do tráfego IP [CISCO 2017].

Dentre as categorias de redes sem fio, as redes móveis *ad hoc*, conhecidas pelo acrônimo MANETs (*Mobile Ad Hoc Networks*), merecem atenção especialmente pelas características de *autoconfiguração*, *autoformação* e *automanutenção* [Wang; Xie; e Agrawal 2009]. Essas redes possuem arquitetura P2P e seus *peers* se comunicam sem depender de qualquer infraestrutura fixa, podendo atuar como servidores, clientes e roteadores. Comunicações em ambientes inóspitos, conferências, aulas e atividades de entretenimento são algumas das suas inúmeras aplicações [Ghalib et al. 2018].

Este contexto é a motivação para o presente artigo, cujo objetivo é propor uma adaptação do BitTorrent para *streaming* de vídeo sob demanda interativo em MANETs. Essa adaptação é denominada de *BitTorrent Interativo para MANET* (BTM-I) e, em linhas gerais, se baseia na ideia de priorizar o compartilhamento de pedaços entre *peers* geograficamente próximos entre si, bem como restringir os pedaços que podem ser requisitados pelos *peers* durante a reprodução do vídeo. Por meio de simulações e avaliando-se diferentes métricas de desempenho, a adaptação BTM-I é aqui validada. Nesse sentido, a principal contribuição deste artigo é prover uma nova solução algorítmica eficiente para o desenvolvimento de novos protocolos para MANETs, com foco em *streaming* sob demanda interativo.

O restante deste artigo é organizado como descrito a seguir. A Seção 2 revisa sucintamente o protocolo BitTorrent e a conceituação geral de redes sem fio, com ênfase em MANETs. A Seção 3 discute trabalhos relacionados. A Seção 4 apresenta a adaptação BTM-I. A Seção 5 traz experimentos, resultados e análises. Por fim, conclusões gerais e caminhos para trabalhos futuros constituem a Seção 6.

2. Fundamentos

2.1. Algoritmo P2P BitTorrent

Como já mencionado, o arquivo que se deseja replicar é dividido em pedaços. Cada pedaço é, por sua vez, dividido em blocos. Os pedaços têm 256 kB de tamanho, enquanto que os blocos têm 16 kB. Apesar de os blocos serem a unidade de dados na rede física, a análise de replicação considera apenas os pedaços transmitidos [Cohen 2003]. Por simplicidade, os termos *peer* e *nó* são doravante utilizados indistintamente

no restante deste texto. A operação do BitTorrent é descrita a seguir, considerando a visão de um *peer p*, recém-chegado ao sistema, que deseja receber um arquivo *A*.

O *peer p* deve inicialmente obter, a partir de um servidor *web* da Internet, um arquivo de metadados relacionado ao arquivo *A*. O arquivo de metadados possui a localização do *tracker*, que é o coordenador da comunicação entre os *peers* que cooperativamente transmitem entre si os pedaços do arquivo *A* pela rede. Esse conjunto de *peers* cooperativos é chamado de *swarm* [Legout; Urvoy-Keller; e Michiardi 2006].

O *tracker*, após contatado pelo *peer p*, envia-lhe uma lista *L*, que contém os *peers* que já estão no *swarm* replicando o arquivo *A*. Essa lista possui tipicamente 50 *peers*. O *peer p* tenta então aleatoriamente estabelecer conexões TCP com os *peers* dessa lista. 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 *A* [Cohen 2003].

Um *leecher* é um *peer* que está baixando pedaços de um arquivo, mas que também permite que outros *peers* baixem pedaços a partir dele. Um *seed* é um *peer* que já tem todos os pedaços do arquivo desejado, mas que permanece no sistema para permitir que outros *peers* baixem pedaços a partir dele. O intercâmbio de pedaços entre os *peers* é operacionalizado pelas políticas de seleção de *peers* e de seleção de pedaços, como explicado a seguir [Legout; Urvoy-Keller; e Michiardi 2006].

A política de seleção de *peers*, também chamada de algoritmo de *choke*, permite que cada *peer p* do *swarm* escolha quais *peers*, dentre os seus vizinhos, podem receber os pedaços que ele possui. Os três vizinhos que fornecem pedaços para o *peer p* com as mais altas taxas de *upload* são os escolhidos. Os *peers* escolhidos são colocados no estado de *unchoked*, e os demais são colocados no estado de *choked*. Um *slot* de *upload* de dados do *peer p* é alocado para cada um dos escolhidos. Essa seleção é repetida tipicamente a cada 10 segundos, sendo denominada de *regular unchoking*. Há ainda o denominado *optimistic unchoking*, conforme explicado a seguir. Tipicamente a cada 30 segundos, cada *peer p* do *swarm* seleciona aleatoriamente um outro *peer* de seu conjunto de vizinhos para poder receber os pedaços que ele possui. Nesse caso, um *slot* de *upload* do *peer p* é então alocado para esse vizinho [Cohen 2003].

A política de seleção de pedaços, por sua vez, é usada para que os *peers* decidam sobre que pedaços solicitar ao passar para o estado de *unchoked*. Cada *peer p* 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. O índice de cada pedaço com *m* cópias no conjunto de vizinhos é então adicionado ao conjunto de pedaços mais raros. Após passar para o estado de *unchoked*, o *peer p* solicita o próximo pedaço considerando seu conjunto de pedaços mais raros e os pedaços disponíveis no vizinho que o passou para o estado *unchoked*. Após receber o pedaço, o *peer p* avisa a todos os seus vizinhos sobre o pedaço recebido [Legout; Urvoy-Keller; e Michiardi 2006].

Como mencionado, apesar de haver adaptações do BitTorrent para *streaming* em redes cabeadas (e.g., [D'Acunto et al. 2013]), a sua aplicação em MANETs ainda é alvo de pesquisa recente (e.g., [Ghalib et al. 2018; Usman et al. 2018]).

2.2. Redes sem Fio

As redes sem fio são basicamente de dois tipos: infraestruturada e sem infraestrutura. As infraestruturadas têm a comunicação entre os nós suportada por uma plataforma fixa e cabeada, responsável por organizar, processar e rotear os dados entre os nós. Essa plataforma possui as *estações bases* como principal elemento de conexão. Sucintamente, todo nó deve se conectar à *estação base* mais próxima dentro de seu raio de alcance. Se este nó for móvel e ocorrer de ele sair da cobertura de sua *estação base* original, então uma outra *estação base*, que estiver dentro de seu raio de alcance, passa a ser a responsável pela sua conexão à rede. O processo de mudança entre *estações bases* é denominado de *handoff*. As redes locais sem fio e as redes celulares comerciais são exemplos típicos [Wang; Xie; e Agrawal 2009].

As redes sem infraestrutura, por sua vez, são caracterizadas por não possuírem uma plataforma fixa e cabeada suportando a comunicação entre nós, os quais são tipicamente móveis. Neste caso, os próprios nós atuam como clientes, roteadores e servidores do sistema, realizando a operacionalização de toda a comunicação. Estas redes são denominadas redes móveis *ad hoc* ou, em inglês, *mobile ad hoc networks* (MANETs). Devido à mobilidade dos nós, a topologia da rede pode mudar rapidamente e de forma imprevisível [Xiang e Yang 2018; Wang; Xie; e Agrawal 2009].

Na MANET, cada nó pode se comunicar diretamente com um outro nó que esteja dentro de seu alcance, i.e., usando um enlace de comunicação de salto único (do inglês, *one-hop wireless link*). Para se comunicar com um nó fora de seu alcance direto, um nó deve utilizar nós intermediários para retransmitir suas mensagens, i.e., deve usar um enlace de comunicação de múltiplos saltos (do inglês, *multiple-hop wireless link*). Como destacado anteriormente, esse tipo de rede permite vislumbrar aplicações em diversas áreas, permeando desde comunicações militares até, por exemplo, atividades de educação e entretenimento em geral [Wang; Xie; e Agrawal 2009].

Com o passar do tempo, a evolução das tecnologias móveis propiciou o surgimento de variações da MANET original, que podem ser admitidas como subcategorias ou como novos tipos independentes. Dentre as mais comuns, citam-se as três seguintes: 1) *Vehicular Ad Hoc Network* (VANET) [Boussoufa-Lahlah; Semchedine; e Bouallouche-Medjkoune 2018; Lobato Junior et al. 2018]; 2) *Flying Ad Hoc Network* (FANET) [Cruz 2018]; e 3) *Wireless Mesh Network* (WMN) [Wang; Xie; e Agrawal 2009]. Para fins de análise desses diversos tipos de redes, destaca-se a importância do uso de modelos de mobilidade [Bai e Helmy 2006].

Para finalizar esta seção, esclarece-se que BTM-I se direciona à MANET original, considerando nós que são indivíduos (estudantes, militares, passageiros, etc.) se movimentando sobre uma área plana, sem obstáculos e sem acesso à Internet. Esse cenário pode representar, e.g., parte de um campus universitário, onde os alunos desejam assistir a uma aula ao ar livre, ou mesmo um saguão de aeroporto, onde os passageiros assistem a um filme antes do embarque. Além disso, o nó pode realizar ações de interatividade (e.g., *pause* e *jump backwards*) durante a reprodução do vídeo.

3. Trabalhos Relacionados

Os trabalhos de Mantzouratos et al. (2012), Shah et al. (2017), Lal, Laxmi e Gaur (2011), e Jahir et al. (2019) se constituem em valiosos *surveys* sobre *streaming* em MANETs. Em específico, o primeiro trata do desenvolvimento de protocolos sob o paradigma *cross layer*, discutindo concepções, estratégias de avaliação, métricas e limitações dos resultados até então alcançados. O segundo possui o mesmo foco geral, diferenciando-se, porém, ao considerar o projeto sob o paradigma *layered architecture*. Por fim, os dois últimos direcionam-se ao roteamento em MANETs, sendo que Jahir et al. (2019) têm uma abordagem mais ampla, incluindo combinações de MANETs e outras redes, resultando em *Disaster Area Networks* (DANs). Embora não sejam apresentadas novas soluções, as constatações feitas nesses trabalhos se tornam relevantes por indicarem caminhos possíveis de serem seguidos ou mesmo evitados.

Quental e Gonçalves (2010) trazem uma das primeiras propostas de extensão do algoritmo BitTorrent para o emprego em MANETs. Seu foco é a distribuição simples de arquivos. A ideia consiste na adição de uma interface entre o núcleo do BitTorrent, executado na camada de aplicação, e a camada de transporte, de forma a permitir o envio e o recebimento de dados via *broadcast*, além da convencional transmissão *unicast*. Em que pese o fato de os resultados obtidos serem satisfatórios para o tempo de *download* do arquivo, não são realizados experimentos para *streaming* sob demanda.

Chen and Shen (2015) apresentam um protocolo de replicação simples de arquivos que considera a capacidade individual de armazenamento do nó móvel, além da periodicidade de encontro entre os nós detentores dos arquivos e os demais nós da rede. São utilizadas simulações para atestar o desempenho do protocolo, mas assim como no trabalho de Quental e Gonçalves (2010), os resultados referem-se apenas ao tempo de *download* do arquivo, não sendo analisado o *streaming* sob demanda.

Hu et al. (2017) propõem um algoritmo de escalonamento de segmentos de vídeo, denominado de *Delay-Sensitive Segment Scheduling Algorithm* (DSSSA). A filosofia é retardar a transmissão dos segmentos cujos instantes de reprodução estão distantes do momento corrente. Isso permite priorizar o uso da banda para aqueles segmentos cuja reprodução deverá ocorrer mais rapidamente. Essa proposta é uma das poucas da literatura a considerar explicitamente limites de tempo para entrega do arquivo, que é essencial para a implementação de *streaming* sob demanda, além de também considerar algum nível de interatividade do cliente em MANETs. São realizadas simulações comparativas com esquemas tradicionais, onde são observadas otimizações na taxa de admissão e na taxa de recepção de dados com sucesso.

Usman et al. (2018) propõem um esquema para *streaming* de vídeos de alta definição em MANETs. Esse esquema é do tipo proativo, usando provedores típicos de serviços, e baseia-se especialmente em estimativas de características de operação da rede como, e.g., taxa de perda de pacotes e atrasos fim a fim. Embora os resultados dos experimentos mostrem uma superioridade de desempenho com relação a protocolos de roteamento específicos de MANETs (e.g., protocolos DSDV e OLSR), não há discussões para um serviço do tipo reativo, caracterizando o *streaming* sob demanda.

Por fim, Rodrigues (2018) propõe dois algoritmos para realização de *streaming* sob demanda em MANETs, tomando como base o algoritmo BitTorrent. A concepção

das propostas avança mudanças na política de seleção de *peers*, que passa a usar uma combinação de quatro critérios, e também mudanças na política de seleção de pedaços, que passa a usar o conceito de janela deslizante para priorização dos pedaços a solicitar. Mas apenas tráfego via canais *unicast* é admitido. Simulações comparativas com um esquema ideal são realizadas e os resultados obtidos enaltecem a eficiência das propostas. Como mencionado para Hu et al. (2017), esse trabalho é um dos poucos da literatura a considerar tempo para entrega do arquivo e admitir a interatividade.

Ante o exposto, a adaptação BTM-I se diferencia das obras anteriores ao serem observadas as seguintes peculiaridades em conjunto: baseia-se no protocolo BitTorrent; aplica-se à *streaming* sob demanda (i.e., reativo); usa uma janela deslizante para priorização dos pedaços do arquivo a serem solicitados; admite tráfego *unicast* e *broadcast* para o compartilhamento dos pedaços do arquivo; o *streaming* ocorre apenas entre vizinhos; usa a distância geográfica como um critério para formação de grupos de vizinhos; explora apenas a camada de aplicação; e, por fim, admite a interatividade dos clientes durante a reprodução do arquivo. A adaptação BTM-I é, assim, independente de protocolos de roteamento, de simples implementação e de relativo ineditismo.

4. Adaptação BTM-I

Esta seção apresenta a adaptação BTM-I. Isso é feito por meio da explicação de suas políticas de seleção de *peers* e de seleção de pedaços, respectivamente, considerando subentendido o arcabouço conceitual da Seção 2.

Para a política de seleção de *peers*, tem-se o seguinte. Seja um *peer* p pertencente à MANET. Os vizinhos do *peer* p são escolhidos aleatoriamente a partir da lista L fornecida pelo *tracker*. Os *peers* escolhidos são colocados no estado *choked* e classificados em *interessados* e *não interessados*: se um vizinho não tem todos os pedaços que o *peer* p possui, este vizinho é classificado como *interessado* no *peer* p ; caso contrário, é classificado como *não interessado* no *peer* p . A banda de *upload* do *peer* p é então dividida em $y+1$ *slots* de dados: y *slots* para *regular unchoking* e 1 *slot* para *optimistic unchoking*. Esses dois processos ocorrem periodicamente até que a reprodução é encerrada. O Quadro 1 traz a descrição desses dois processos.

Para a política de seleção de pedaços, tem-se o seguinte. Admita que o arquivo de vídeo é dividido em t pedaços: 1, 2, ..., t . Seja d o pedaço que corresponde ao ponto de reprodução corrente. Seja W a janela deslizante, e seja w o seu tamanho em número de pedaços. A janela W compreende então os seguintes pedaços: $[d; d + w]$. A janela W é dinamicamente atualizada, conforme os pedaços são reproduzidos e as ações de interatividade são realizadas. Por exemplo, seja j_{play} o número de pedaços reproduzidos. O primeiro pedaço de W é atualizado para $(d + j_{\text{play}})$, e o último pedaço para $(d + w + j_{\text{play}})$. Seja agora V um *buffer* contido em W , e seja v o seu tamanho em número de pedaços. O primeiro pedaço de V é sempre coincidente com o primeiro pedaço de W . Antes de o *peer* p solicitar o próximo pedaço, é verificado se o seu *buffer* V está cheio. Em caso positivo, solicita-se o pedaço mais raro considerando a janela W ; caso contrário, solicita-se de maneira gulosa (i.e., sequencial), o próximo pedaço faltante do *buffer* V . A alternância de método para escolher o próximo pedaço busca estabelecer um compromisso entre a diversidade de pedaços, estimulada pela política do pedaço

mais raro, e a continuidade na reprodução do arquivo, estimulada pela política gulosa, como deve ser em um sistema de *streaming* sob demanda.

Regular unchoking – a cada 10 segundos, tem-se:

Se o *peer p* for um *leecher*:

Os *peers interessados* são ordenados em função de suas taxas de *upload* de dados. Dentre os x *peers* de maiores taxas, os y *peers* geograficamente mais próximos (i.e., menor distância) são escolhidos e aloca-se 1 *slot* para cada um deles, com $x \geq y$. Em caso de empate, a escolha é aleatória.

Se o *peer p* for um *seed*:

Os *peers interessados* são ordenados em função da taxa de *download* de dados a partir do *peer p*. Dentre os x *peers* de maiores taxas, os y *peers* fisicamente mais próximos (i.e., menor distância) são escolhidos e aloca-se 1 *slot* a cada um deles, com $x \geq y$. Em caso de empate, a escolha é aleatória. Se for a primeira vez que o pedaço é solicitado, então a transmissão é em *broadcast*; caso contrário em *unicast*.

Optimistic unchoking – a cada 30 segundos, tem-se:

Conforme proposta original do protocolo BitTorrent: 1 *slot* de dados é alocado a um *peer interessado* escolhido aleatoriamente.

Quadro 1. Seleção de *peers* do BTM-I.

5. Avaliação de Performance

5.1. Modelagem, Cenários e Métricas

Os experimentos são implementados no ambiente de simulação PeerSim [Montresor e Jelasity 2009], o qual se constitui em uma ferramenta confiável para realização de simulações por sua ampla aceitação na literatura. A plataforma computacional é um PC de processador Intel i7, 2,6 GHz de *clock*, 24 GB de RAM, e sistema GNU/Linux. Os resultados de simulação têm intervalos de confiança de 95% que estão dentro do limite de 5% dos valores médios reportados, tendo sido realizadas 10 execuções (rodadas). Ressalta-se que, por restrição de espaço, apenas os resultados mais relevantes para suporte às conclusões atingidas são aqui apresentados.

Os cenários simulados possuem n nós que se movimentam de acordo com padrões de mobilidade de indivíduos gerados pelo simulador *Small World in Motion* (SWIM) [Mei e Stefa 2009]. Esses padrões são fundamentados no entendimento de que um indivíduo tende a visitar lugares populares ou próximos ao lugar onde permanece a maior parte do tempo (e.g., escola, cafeteria, saguão, etc.), revelando satisfatória aderência ao propósito de aplicação da proposta BTM-I. Além disso, ressalta-se que o modelo de mobilidade SWIM é uma das alternativas consideradas pelo conhecido *framework* de produção de cenários de mobilidade *BonnMotion*.

Na camada de rede, é utilizado o protocolo de roteamento de pacotes baseado no AODV do simulador WiFi Direct [Baresi; Derakhshan; e Guinea 2016]. Para as camadas MAC e física, são considerados os protocolos CSMA/CA e 802.11, respectivamente, com um percentual de perda de pacotes de 1,0%. A largura de banda do nó é de 4 Mbps por *slot* (*upload* e *download*).

O arquivo distribuído por *streaming* sob demanda corresponde a um vídeo com codificação de 300 kbps. O vídeo é dividido em 390 pedaços de 256 kB, em que cada pedaço é dividido em blocos de 16 kB, totalizando aproximadamente 100 MB. O *swarm* é composto por 100 nós, que inicializam suas atividades (entram na rede) ao mesmo tempo no começo da simulação.

As ações de interatividade executadas pelos nós podem ser: *Play*, *Pause*, *Jump Forwards (JF)* e *Jump Backwards (JB)*. *Play* indica que o nó reproduz o arquivo. *Pause* indica que a reprodução é congelada. *JF* e *JB* indicam um salto para um ponto de reprodução à frente e atrás do corrente, respectivamente. A reprodução em cada nó é encerrada quando este nó baixa o arquivo completo.

Nos experimentos são avaliadas quatro métricas de desempenho: (i) *download operacional*, que corresponde à taxa média de um nó para receber o vídeo; (ii) *tempo de download*, que corresponde ao tempo médio de um nó para receber o vídeo; (iii) *tempo de descontinuidade*, que corresponde ao tempo médio total de interrupção de um nó durante a reprodução do vídeo; (iv) *descontinuidades*, que corresponde ao número médio total de pedaços indisponíveis de um nó durante a reprodução do vídeo.

5.2. Organização dos Experimentos

São realizados seis experimentos. Para comparação com a proposta BTM-I, aqui denominada por versão *manet*, também são implementadas a versão *cabeada* e a proposta AD-RI (Algoritmo de Distância Mínima com Reciprocidade Indireta) [Rodrigues 2018]. A versão *cabeada* difere da versão *manet* por considerar uma rede com fio, possuindo nós sem mobilidade e fazendo uso do protocolo de roteamento OSPF. A versão *cabeada* serve como um referencial teórico ideal de desempenho. Por sua vez, AD-RI é bem recente e de mesmo propósito de aplicação que BTM-I. Além da taxa de *upload*, a seleção de *peers* em AD-RI considera a distância geográfica e o compartilhamento indireto de dados entre os nós. Ressalta-se que alguns parâmetros de configuração utilizados têm seus valores provenientes de trabalhos anteriores da literatura ou de análises não apresentadas aqui que observam limites adequados de operação. Não se detalha aqui a determinação desses valores por restrição de espaço.

Para os primeiros quatro experimentos, tem-se que o tamanho do *buffer* V da versão *manet* varia desde $v = 10$ pedaços ($\approx 2\%$ do arquivo) até $v = 120$ pedaços ($\approx 30\%$ do arquivo), e a janela W é fixada em $w = 390$ pedaços, correspondendo ao total do arquivo. Para a versão *cabeada*, o valor do *buffer* V é fixado em $v = 30$ pedaços. Uma métrica distinta é avaliada em cada um desses quatro experimentos. O quinto e o sexto experimentos avaliam a escalabilidade das propostas BTM-I e AD-RI, respectivamente, ante o aumento do número de nós na rede. Nos seis experimentos, os valores dos parâmetros de configuração x e y (vide Quadro 1) são iguais a 10 e 3, respectivamente.

Em relação à mobilidade, os valores utilizados no SWIM para os cinco experimentos são os mesmos utilizados no trabalho de Mei e Stefa (2009), com um tempo máximo de contato (i.e., máxima duração do encontro) entre os nós de 300 segundos. Nesse cenário, os n nós se movimentam sobre uma área plana A , sem obstáculos, de 100 m x 100 m. A velocidade de cada nó é escolhida entre 0 e 2 m/s, e a sua direção entre 0 e 360 graus. Dentre esses n nós, há um *tracker* e um *seed* que operam ininterruptamente e são diretamente alcançáveis por todos os outros nós.

Sobre a interatividade, são considerados três perfis: *interatividade baixa (IB)*, *interatividade média (IM)* e *interatividade alta (IA)*. As ações *Play*, *Pause*, *JF* e *JB* são disparadas segundo uma distribuição de *Poisson* de taxa λ , e probabilidades associadas de p_0 , p_1 , p_2 e p_3 , respectivamente. *Play* possui um intervalo l_{play} de duração. *Pause* possui um intervalo l_{pause} de duração. *JF* e *JB* são instantâneas, possuindo ambas um salto de tamanho l_{jump} . Os valores de l_{play} , l_{pause} e l_{jump} são calculados como um percentual do tamanho total do vídeo f_{size} . A Tabela 1 traz os valores desses parâmetros de configuração, os quais são os mesmos utilizados no trabalho de Rodrigues (2018).

5.3. Resultados e Análises

Na Figura 1(a), tem-se o experimento relativo à métrica *download operacional*. A versão *cabeada* (não mostrada na figura) tem o melhor desempenho nesse experimento, alcançando ≈ 1.300 kbps nos três perfis. Por sua vez, a versão *manet* tem seu melhor desempenho para um *buffer* de tamanho $v = 30$ nos três perfis: 381 kbps (*IB*), 397 kbps (*IM*), e 411 kbps (*IA*). Embora a versão *manet* tenha desempenho abaixo do patamar teórico ideal (i.e., versão *cabeada*), os resultados absolutos permitem a visualização do vídeo com atrativa qualidade de serviço (QoS) em todos os perfis, pois os valores registrados estão acima do valor de sua codificação, i.e., 300 kbps.

Tabela 1. Perfis de interatividade

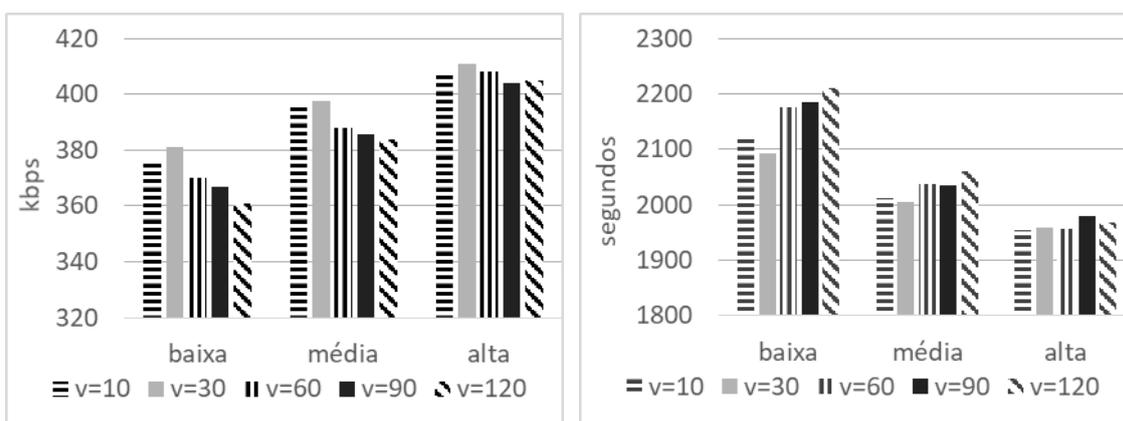
Parâmetro	Baixa (<i>IB</i>)	Média (<i>IM</i>)	Alta (<i>IA</i>)
λ	0,005/s	0,014/s	0,025/s
l_{play}	14,5% de f_{size}	3,5% de f_{size}	1,5% de f_{size}
l_{jump}			
l_{pause}			
$p_0; p_1; p_2; p_3$	0,89;0,01;0,05;0,05	0,71;0,05;0,12;0,12	0,55;0,15;0,15;0,15

É interessante observar que, conforme o valor de v aumenta (i.e., $v > 30$) ou diminui (i.e., $v < 30$), o desempenho se degenera progressivamente. Com respeito a essa perda de desempenho, conjectura-se a razão a seguir. Para $v < 30$, ocorre a situação de um *buffer* pequeno que é rapidamente consumido durante a reprodução. Isso resulta em frequentes indisponibilidades de pedaços localmente, o que aumenta a disputa por pedaços no *swarm*, comprometendo a taxa de *download* de cada nó. Para $v > 30$, tem-se a condição de um *buffer* grande que nunca é preenchido. Isso faz com que a recuperação de pedaços seja estritamente sequencial. Neste caso, a diversidade de pedaços no *swarm*, promovida pela recuperação dos pedaços mais raros, deixa de existir, comprometendo a disponibilidade de pedaços no *swarm* e, conseqüentemente, a taxa de *download* de cada nó. Assim, quanto maior (ou menor) for o tamanho do *buffer*, com relação ao valor de referência $v = 30$, pior tende a ser o desempenho.

A Figura 1(b) traz o experimento para a métrica *tempo de download*. A versão *cabeada* (não mostrada na figura) tem novamente o melhor desempenho: ≈ 588 s para todos os perfis de interatividade. Por sua vez, para a versão *manet*, o melhor desempenho é novamente alcançado para um *buffer* de tamanho $v = 30$ em todos os perfis, obtendo-se: 2092 s (*IB*), 2006 s (*IM*), e 1959 s (*IA*). Esses resultados ratificam as observações e conjecturas da discussão anterior sobre a métrica *download operacional*.

Na Figura 2(a), podem ser vistos os resultados dos experimentos da métrica *tempo de descontinuidade*. Como esperado, a versão *cabeada* mais uma vez supera a versão *manet* em todos os perfis de interatividade. No entanto, diferentemente dos dois experimentos anteriores, a otimização do desempenho da versão *manet* não é mais obtida para um mesmo tamanho de *buffer*: 40 s ($v = 90$ para *IB*); 229 s ($v = 30$ para *IM*); e 335 s ($v = 30$ para *IA*). Esses valores correspondem a apenas 1,8%, 11,4% e 17,1% dos *tempos de download* para os perfis *IB*, *IM* e *IA*, respectivamente. Não obstante, se necessário, pode-se buscar mitigar esses valores por meio de técnicas como *buffering*, *prefetching*, intercalação e/ou *Forward Error Correction* (FEC).

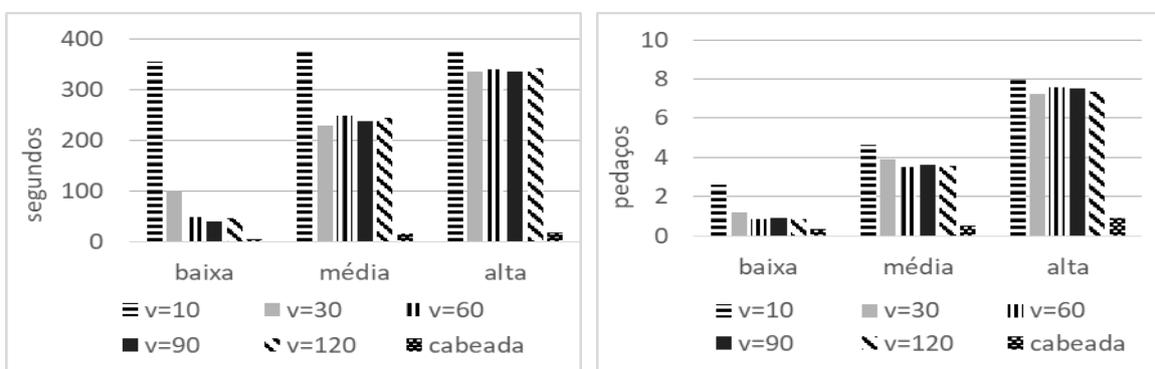
A Figura 2(b) traz a análise da métrica *descontinuidades*. A versão *cabeada* novamente supera a versão *manet* em todos os perfis. Como ocorrido no experimento anterior, a versão *manet* também registra seu melhor desempenho com tamanhos de *buffer* diferentes: 0,9 pedaço ($v = 60$ para *IB*); 3,5 pedaços ($v = 60$ para *IM*); e 7,2 pedaços ($v = 30$ para *IA*). Entretanto, note que, para cada perfil, os pedaços indisponíveis para *buffers* com valores de v entre 30 e 120 não apresentam uma diferença significativa, não chegando a 1,0 pedaço.



(a) *Download operacional.*

(b) *Tempo de download.*

Figura 1. Download operacional e tempo de download da versão *manet*.



(a) *Tempo de descontinuidade.*

(b) *Descontinuidades.*

Figura 2. Tempo de descontinuidade e descontinuidades na reprodução.

Embora o valor ideal do *buffer* não tenha se mantido idêntico nos três perfis de interatividade nos experimentos ilustrados nas Figuras 2(a) e 2(b), como ocorreu nos dois primeiros experimentos, o valor de v que otimiza o desempenho da versão *manet* nesses experimentos ainda continua sendo um ponto de referência: a partir do qual (valores maiores ou menores), pode-se observar uma tendência de perda progressiva de desempenho. Note ainda que os resultados desses dois últimos experimentos estão em acordo com as observações e conjecturas dos dois primeiros experimentos.

A partir dos resultados e discussões tidas até este momento, é possível concluir que o algoritmo BTM-I (denotado como versão *manet*) tem performance promissora. Isso é justificado porque o sistema é capaz de prover uma taxa de recuperação do arquivo maior que a taxa de codificação do mesmo (e.g., vide Figura 1) e, também, as discontinuidades medidas não são significativas (e.g., vide Figura 2), estando em patamares aceitáveis para um típico serviço de *streaming* sob demanda interativo, ainda mais especialmente quando a interatividade é baixa. Além disso, também pode ser concluído que o tamanho do *buffer* V pode ser estimado, na maioria dos casos, em 8% (i.e., $v = 30$ pedaços) do tamanho do arquivo, pois o desempenho é semelhante, ou mesmo melhor em alguns casos, ao de *buffers* de tamanhos maiores ou menores. Sobre a superioridade da versão *cabeada*, é possível então concluir que há espaço para a otimização da versão *manet*, o que pode ser alcançado pela adição de outros critérios de seleção de pedaços ou de *peers*, sendo isso uma investigação para trabalhos futuros.

Na Figura 3, pode-se observar os resultados do quinto experimento, o qual trata da escalabilidade da proposta BTM-I. O valor do *buffer* V é fixado em $v = 30$, com a janela W fixada em $w = 390$. Os resultados mostram que o aumento de nós, variando de 25 até 100, não tem impacto significativo sobre as métricas. Por exemplo, para *download operacional*, no perfil *IB*, o maior e o menor valor são 402 e 381 kbps, respectivamente, correspondendo a uma diferença de apenas 5,2%. Para *tempo de download*, no mesmo perfil (*IB*), a diferença percentual entre o maior e o menor valor corresponde a apenas 5,3%. Por fim, para *tempo de descontinuidade* e *descontinuidades*, também no perfil (*IB*), a diferença percentual entre os maiores e o menores valores associados é de $\approx 30,0\%$, correspondendo a apenas 30 s e ≈ 1 pedaço, respectivamente. Isso permite, portanto, concluir positivamente sobre a escalabilidade.

Por fim, a Figura 4 traz o sexto e último experimento, o qual examina a escalabilidade da proposta AD-RI. Por restrição de espaço, são ilustradas apenas as métricas *download operacional* e *tempo de download*. Os resultados permitem concluir que AD-RI é escalável, pois o aumento do número de nós não influencia significativamente as métricas. Comparando-se BTM-I à AD-RI, os resultados mostram o seguinte em todos os perfis de interatividade. Para *download operacional* (Figuras 3(a) e 4(a)), BTM-I provê um ganho mínimo de $\approx 20\%$ (i.e., 80 kbps). Para *tempo de download* (Figuras 3(b) e 4(b)), BTM-I provê uma redução mínima de $\approx 22\%$ (i.e., 500 s). Para *tempo de descontinuidade*, BTM-I provê uma redução mínima de $\approx 20\%$ (i.e., 20 s). Para *descontinuidades*, BTM-I provê uma redução mínima de $\approx 50\%$ (i.e., 2 pedaços). Em síntese, BTM-I é superior à AD-RI. Conjectura-se que essa superioridade ocorre por BTM-I considerar transmissão *broadcast*, implicando maior efetividade no compartilhamento de dados, e ter uma política de seleção de *peers* mais simples, implicando menor *overhead* de operação.

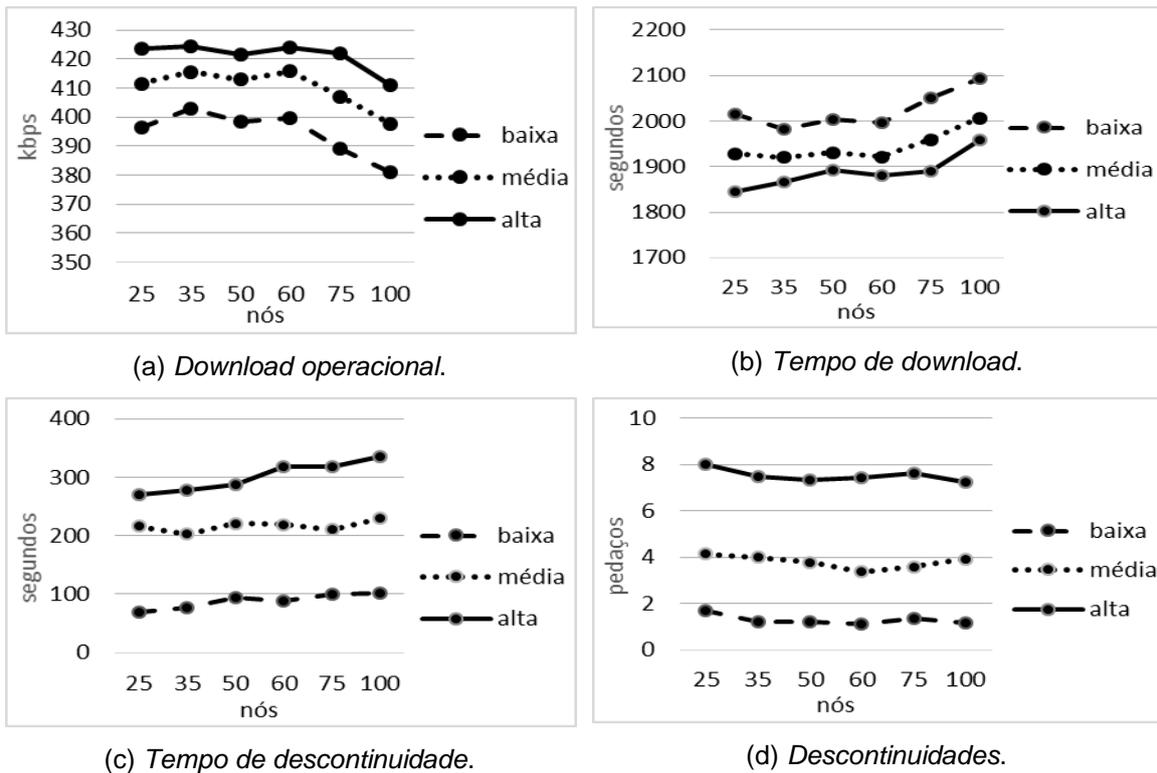


Figura 3. Escalabilidade de BTM-I.

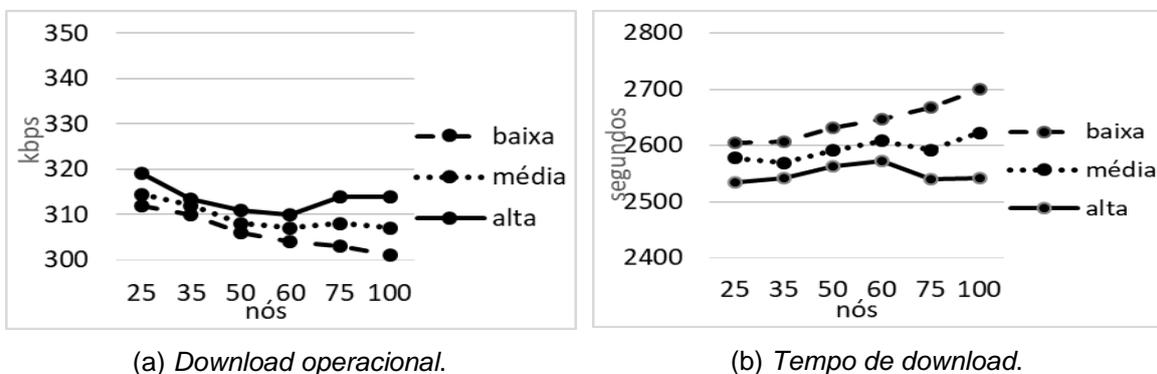


Figura 4. Escalabilidade de AD-RI.

6. Conclusões e Trabalhos Futuros

Este artigo apresentou uma adaptação do algoritmo P2P BitTorrent, denominada de BTM-I, para a realização de *streaming* sob demanda interativo em MANETs. Por meio de simulações e considerando quatro distintas métricas de desempenho, constatou-se a eficiência da proposta e concluiu-se positivamente sobre a QoS do sistema.

Como resultados pontuais, destacam-se: a taxa de recuperação de dados e a continuidade de reprodução ficaram em patamares atrativos; a escalabilidade da adaptação BTM-I foi confirmada considerando MANETs de até 100 nós; por fim, o envio de dados em *broadcast*, o uso de uma janela para priorização dos dados a

recuperar, e o uso da distância geográfica entre os nós para seleção de *peers* se revelaram ideias promissoras para o projeto de protocolos de *streaming* em MANETs.

Para trabalhos futuros, apontam-se: complementar esta análise com modelos analíticos e medições reais, incluindo peculiaridades das camadas inferiores (i.e., MAC e física) [Silva e Moraes 2018]; realizar comparações com outras propostas da literatura ante: distintas áreas de cobertura, variados modelos de mobilidade e interatividade [Domingues et al. 2018], nós heterogêneos e múltiplas codificações de arquivo [Mantzouratos et al. 2012]; por último, avaliar distintos roteamentos [Melo et al. 2017].

Referências

- Bai, F. and Helmy, A. (2006). A Survey of mobility models in wireless ad hoc network. In: (eds.) *Wireless Ad Hoc and Sensor Networks*, Springer, October, pp. 1–30. Available at: <https://www.cise.ufl.edu/~helmy/papers/Survey-Mobility-Chapter-1.pdf>. Accessed on: Nov. 17th, 2018.
- Baresi, L.; Derakhshan, N.; and Guinea, S. (2016). WiDiSi: A Wi-Fi direct simulator. In: *IEEE Wireless Communications and Networking Conference*, Doha, Qatar.
- Boussoufa-Lahlah, S.; Semchedine, F.; and Bouallouche-Medjkoune, L. (2018). Geographic routing protocols for Vehicular Ad hoc NETWORKS (VANETs): A survey. *Vehicular Communications*, v. 11, January, pp. 20–31.
- Chen, K. and Shen, H. (2015). Maximizing P2P file access availability in mobile ad hoc networks through replication for efficient file sharing. *IEEE Transactions on Computers*, v. 64, n. 4, pp. 1029–1042.
- CISCO. (2017). Cisco Visual Networking Index: Forecast and Methodology, 2016–2021. Available at: <https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf>. Accessed on: Nov. 12th, 2018.
- Cohen, B. (2003). Incentives build robustness in BitTorrent. In: *First Workshop on Economics of Peer-to-Peer Systems*, Berkeley, USA.
- Cruz, E. P. F. (2018). A Comprehensive Survey in Towards to Future FANETs. *IEEE Latin America Transactions*, v. 16, n. 3, pp. 876–884.
- D’Acunto, L. et al. (2013). BitTorrent-like P2P approaches for VoD: A comparative study. *Computer Networks*, v. 57, n. 5, pp. 1253–1276.
- Domingues, A. et al. (2018) Perfil de Uso de Aplicativos Móveis: Caracterização e Aplicações. In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2018)*, Campos do Jordão, SP, Brasil.
- Ghalib, S. et al. (2018). Routing protocol development for quality of service optimization of video-on-demand system over mobile ad hoc networks. *International Journal of Communications Systems*, v. 31, n. 2.
- Hu, C.-C. et al. (2017). Timely scheduling algorithm for P2P streaming over MANETs. *Computer Networks*, v. 127, pp. 56–67.

- Jahir, Y. et al. (2019). Routing protocols and architecture for disaster area network: A survey. *Ad Hoc Networks*, v. 82, pp. 1–14.
- Lal, C.; Laxmi, V.; and Gaur, M. S. (2011). Performance Analysis of MANET Routing Protocols for Multimedia Traffic. In: *International Conference on Computer & Communication Technology (ICCCT)*, Allahabad, India.
- Legout, A.; Urvoy-Keller, G; and Michiardi, P. (2006). Rarest first and choke algorithms are enough. In: *6th ACM SIGCOM Conference on Internet Measurement*, Rio de Janeiro, RJ, Brazil.
- Lobato Junior, W. et al. (2017). Mecanismo para Cooperação e Coligação de Veículos Baseado na Teoria dos Jogos para Transmissão de Vídeos em VANETs. In: *16º Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance 2017)*, São Paulo, SP, Brasil.
- Mantzouratos, S. et al. (2012). Survey of Cross-layer Proposals for Video Streaming over Mobile Ad hoc Networks (MANETs). In: *International Conference on Telecommunications and Multimedia (TEMU)*, July.
- Mei, A. and Stefa, J. (2009). SWIM: A Simple Model to Generate Small Mobile Worlds. In: *IEEE INFOCOM 2009*, Rio de Janeiro, RJ, Brazil.
- Melo, R. et al. Remote Routing Approach to Restricted Devices in MANETs. (2017). In: *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC 2017)*, Belém, Pará, Brasil.
- Montresor, A. and Jelasity, M. (2009). PeerSim: A scalable P2P simulator. In: *9th International Conference on Peer-to-Peer (P2P'09)*, Seattle, Washington, USA.
- Quental, N. and P. Gonçalves, P. (2010). Mobile-BitTorrent: a BitTorrent Extension for MANETs. *Revista de Exatas e Tecnológicas*, v. 1, n. 1.
- Rodrigues, C. K. S. (2018). Efficient BitTorrent-Like Algorithms for Interactive On-Demand Multimedia Streaming over MANETs. In: *24th Brazilian Symposium on Multimedia and the Web (WebMedia'18)*, Salvador, BA, Brazil.
- Shah, N. et al. (2017). A survey of P2P content Sharing in MANETs. *Computers and Electrical Engineering*, v. 57, pp. 55–68.
- Silva, B. e Moraes, R. (2018). Consumo de Energia em Função da Taxa de Transmissão e do Tamanho do Pacote em Redes Ad Hoc Aloha de um Salto. In: *17º Workshop em Desempenho de Sistemas Computacionais e de Comunicação (WPerformance 2018)*, Natal, RN, Brasil.
- Usman, M. et al. (2018). Performance evaluation of High Definition video streaming over Mobile Ad Hoc Networks. *Signal Processing*, v. 148, pp. 303–313.
- Wang, J; Xie, B.; and Agrawal, D. P. (2009). Journey from Mobile Ad Hoc Networks to Wireless Mesh Networks. In: (eds) *Guide to Wireless Mesh Networks*. Computer Communications and Networks. Springer, London.
- Xiang, S. and Yang, J. (2018). Performance reliability evaluation for mobile ad hoc networks. *Reliability Engineering & System Safety*, v. 169, pp. 32–39.