

CDS-BitTorrent: Um Sistema de Disseminação de Conteúdo para a Melhoria do Desempenho de Aplicações BitTorrent sobre MANETs

Nivia Cruz Quental, Paulo André da S. Gonçalves

Centro de Informática (CIn)
Universidade Federal de Pernambuco (UFPE)
50.740-540 – Recife – PE – Brasil

{ncq, pasg}@cin.ufpe.br

Abstract. *This paper introduces a content dissemination system, namely, CDS-BitTorrent. This system extends BitTorrent in order to improve download performance in MANETs that are composed by mobile devices. To improve performance, CDS-BitTorrent adopts specific strategies for selecting some content and disseminating it through a limited broadcast. Simulation results obtained by using NS-2 indicate that CDS-BitTorrent is able to reduce the download time by up to 30% and the overhead of segments by up to 50% when compared to the traditional BitTorrent approach and in accordance with the scenarios studied.*

Resumo. *Este artigo introduz o CDS-BitTorrent como um sistema de disseminação de conteúdo que estende o BitTorrent para a melhoria de seu desempenho de download em MANETs formadas por dispositivos móveis. Para alcançar tal objetivo, o CDS-BitTorrent adota estratégias específicas de seleção de parte do conteúdo e de disseminação dessa parte através de um broadcast controlado. Resultados de simulação com o NS-2 mostram que o CDS-BitTorrent permite uma redução de até 30% no tempo de download e uma redução de até 50% no overhead de segmentos enviados, quando comparado ao BitTorrent tradicional e de acordo com os cenários estudados.*

1. Introdução

As MANETs (*Mobile Ad Hoc Networks*) são redes móveis que possuem como características a autonomia dos nós, a auto-organização, o dinamismo da topologia, o roteamento em múltiplos saltos e a formação em caráter tipicamente temporário. Todas essas características também estão presentes na maioria das redes P2P (*Peer-to-Peer*). Tais semelhanças entre MANETs e redes P2P têm culminado em pesquisas que buscam explorar suas sinergias e melhorar o desempenho de aplicações P2P onde a adoção de redes sem fio se faz necessária [Hu et al. 2005].

Em particular às abordagens P2P existentes, o protocolo BitTorrent [Cohen 2006] tem chamado a atenção da comunidade de pesquisa por sua eficiência no compartilhamento de arquivos extensos na Internet. Trazer tal eficiência para ambientes de MANETs é um grande desafio de pesquisa que, por sua vez, vem incentivando estudos recentes que buscam soluções para prover um funcionamento eficiente do protocolo BitTorrent, considerando as características das MANETs e dos cenários de uso dessas redes.

Este artigo introduz um sistema de disseminação de conteúdo, denominado CDS-BitTorrent (*Content Dissemination System - BitTorrent*). Ele estende o protocolo BitTorrent com o objetivo de melhorar o desempenho de seu processo de *download* em um ambiente de MANETs, usando estratégias específicas de seleção de parte do conteúdo e de disseminação dessa parte através de um *broadcast* controlado. O grande diferencial do CDS-BitTorrent é a busca por uma solução totalmente focada na camada aplicação, permitindo que usuários de dispositivos móveis, como *smartphones* e PDAs, se beneficiem de melhorias de desempenho através da adoção de um sistema que não requer alterações de *firmware*. Este artigo avalia as ideias de seleção e disseminação de conteúdo introduzidas pelo CDS-BitTorrent, focando em cenários onde há, inicialmente, apenas uma fonte de conteúdo (como em conferências e palestras) e deseja-se enviar arquivos extensos em um menor espaço de tempo possível para os *peers* da rede.

O restante deste artigo está organizado como segue: a Seção 2 apresenta os trabalhos relacionados à adaptação do BitTorrent em MANETs. A Seção 3 detalha o protocolo BitTorrent. A Seção 4 é dedicada à apresentação do CDS-BitTorrent. A Seção 5 apresenta os cenários de simulação e os resultados de avaliação de desempenho do BitTorrent tradicional e do CDS-BitTorrent. Finalmente, a Seção 6 apresenta as conclusões deste trabalho.

2. Trabalhos Relacionados

A melhoria do desempenho do BitTorrent em cenários de MANETs é um tópico importante que vem sendo abordado recentemente em diversas pesquisas [Rajagopalan and Shen 2006] [Krifa et al. 2009] [Souza and Nogueira 2008]. Essas propostas são apresentadas a seguir, enfatizando-se como as mesmas se diferenciam do CDS-BitTorrent.

Rajagopalan *et al.* [Rajagopalan and Shen 2006] buscam uma maior eficiência do BitTorrent em MANETs com o uso de técnicas de *cross-layering* ao se integrar funcionalidades da camada aplicação com uma camada de roteamento baseada no protocolo denominado ANSI.

Krifa *et al.* [Krifa et al. 2009] propõem o BitHoc, uma solução *cross-layer* para BitTorrent em redes *ad hoc*. A solução se divide em um componente de gerenciamento de *peers* no papel de *Tracker* e um componente de compartilhamento de conteúdo, adaptando a seleção de peças de acordo com a topologia da MANET. O BitHoc conta com informações provenientes do protocolo de roteamento OLSR (*Optimized Link State Routing*) para atualizar o Tracker sobre os *peers* disponíveis.

Em [Souza and Nogueira 2008], é proposto um conjunto de modificações no protocolo BitTorrent para lidar com a localidade espaço-temporal, uma propriedade presente em redes *ad hoc* críticas, nas quais nós próximos tendem a baixar o mesmo conteúdo ao mesmo tempo. A proposta apresentada explora recursos de diversas camadas, em uma abordagem *cross-layering*, e consiste em agregar os *peers* da rede em *clusters*, cada qual com seu líder. O líder é responsável por fazer o *download* do conteúdo e por repassá-lo em *multicast* aos *peers* de seu *cluster*. Esses *peers* são desconectados da rede BitTorrent e aguardam passivamente pelos conteúdos do líder.

O CDS-BitTorrent se difere dos trabalhos relacionados por focar apenas na camada aplicação para melhoria do desempenho de *download*. Assim, o CDS-BitTorrent

não interfere na arquitetura da pilha de protocolos e é independente da escolha do protocolo de roteamento da MANET. Uma importante consequência é permitir que usuários de dispositivos móveis, principalmente em MANETs formadas por *smartphones* e PDAs, se beneficiem de melhorias de desempenho sem a necessidade de alteração do *firmware* desses dispositivos. Atualmente, o *firmware* da grande maioria desses dispositivos é fechado e não permite, por parte dos usuários, alterações de camadas inferiores à camada aplicação. Adicionalmente, as soluções puramente baseadas na camada aplicação não dependem de políticas específicas de suporte por parte dos fabricantes e facilita uma adoção maior e mais rápida por parte dos usuários.

3. O Protocolo P2P BitTorrent

O BitTorrent [Cohen 2006] é atualmente um dos principais protocolos P2P adotados na Internet para a obtenção de arquivos extensos. Sua arquitetura híbrida mistura a simplicidade obtida com o uso de entidades centralizadas e a flexibilidade garantida pela autonomia dos *peers* da rede.

No BitTorrent, a rede faz uso de uma entidade denominada *Tracker*, a qual é responsável por responder a consultas feitas via HTTP por parte daqueles que desejam ter informações sobre quais *peers* possuem interesse no mesmo arquivo. Os dados para a consulta são obtidos a partir de um arquivo `.torrent` que descreve o conteúdo desejado e fornece a URL do *Tracker*. Uma vez obtida essa informação, a comunicação com o *Tracker* não é mais obrigatória e é possível a comunicação direta entre os *peers* por meio de um protocolo específico. Ocasionalmente, a comunicação com o *Tracker* pode ser retomada para atualização de informações.

A garantia do equilíbrio das cooperações é uma característica marcante do BitTorrent. Os *peers* que mais colaboram conseguem obter conteúdo mais rapidamente, incentivando o compartilhamento entre usuários. Isso é possível graças ao mecanismo de *choking*, o qual permite que alguns *uploads* sejam bloqueados temporariamente para evitar a degradação do desempenho daqueles que mais contribuem na rede.

A eficiência no *download* de arquivos extensos se deve à forma como eles podem ser fracionados, permitindo que a obtenção dos mesmos seja feita a partir de múltiplas fontes simultaneamente, aumentando a tolerância a falhas do sistema. Um arquivo é dividido logicamente em frações denominadas *pieces*, que são subdivididas em blocos. Tanto os *pieces* quanto os blocos possuem tamanho fixo, cujo valor depende da implementação. Um *piece* é identificado por um índice que define a ordem que o mesmo ocupa no arquivo. Já o bloco é identificado pelo seu *offset*, ou seja, pelo seu deslocamento em *bytes* dentro de um *piece*. Ao longo do *download*, os *peers* fazem requisições por blocos e quando um *peer* completa um *piece*, ele informa aos demais sobre sua nova aquisição.

Na implementação clássica do BitTorrent, prioriza-se a requisição de *pieces* mais raros ainda não obtidos. Já o bloco, é escolhido aleatoriamente entre os que ainda precisam ser obtidos [Cohen 2006]. Um *peer* que possui apenas parte do arquivo é denominado *leecher* enquanto o *peer* que possui o arquivo completo é chamado de *semente*. O processo de *download* e as mensagens do protocolo são detalhados nas próximas seções.

3.1. Processo de *download*

O processo de *download* de um arquivo, por parte de um *peer* interessado, ocorre de acordo com os seguintes passos [Rajagopalan and Shen 2006]:

1. O *peer* faz uma consulta ao *Tracker*, por meio do protocolo HTTP, usando os parâmetros contidos no arquivo `.torrent`. Esse arquivo é previamente criado e disseminado por uma semente inicial;
2. O *Tracker* responde com uma lista de *leechers* e/ou sementes em processo de *download/upload* do arquivo naquele momento;
3. O *peer* se torna parte desse conjunto de *peers* interessados no arquivo e tem condições de trocar mensagens com os demais, tanto para enviar quanto para receber blocos do arquivo;
4. Durante a transferência, os *peers* podem trocar *bitfields*; o *bitfield* é uma informação que indica os *pieces* que ainda não foram baixados de um arquivo em um *peer* particular; cada bit dessa informação está relacionado a um *piece*;
5. Ao fim do *download*, o *peer* pode decidir se tornar uma semente do arquivo. Se decidir se tornar semente, precisará informar o *Tracker* sobre isso.

3.2. Mensagens do Protocolo

O diálogo entre os *peers* da rede BitTorrent é feito por meio de um protocolo próprio que permite fazer requisições por blocos, bloquear o *upload* para algum *peer* e compartilhar informações com os demais *peers* sobre o andamento do *download*. Na Internet, as implementações desse protocolo contam tipicamente com o TCP na camada de transporte [Cohen 2006]. As mensagens do BitTorrent são descritas a seguir:

- HANDSHAKE - usada para o primeiro contato entre 2 *peers*;
- BITFIELD - carrega informação sobre *pieces* já baixados, ou seja, de *bitfield*;
- INTERESTED - informa que um *peer* está interessado nos *pieces* de outro *peer*;
- REQUEST - usada para requisitar um *piece*, especificando o *offset* do bloco desejado;
- PIECE - traz consigo um bloco de um *piece*, informando o índice do *piece* e o *offset* do bloco;
- HAVE - informa que um *peer* acabou de completar um *piece*;
- CHOKE - informa a um *peer* que este está impedido de requisitar *pieces* de um outro *peer*;
- UNCHOKE - informa a um *peer* que este pode requisitar *pieces* de um outro *peer*.

4. O Sistema de Disseminação CDS-BitTorrent

O sistema de disseminação CDS-BitTorrent introduz estratégias específicas de seleção e disseminação de conteúdo para a melhoria do desempenho do processo de *download*. Tal sistema permite que uma pequena parcela do conteúdo seja entregue através de um *broadcast* controlado, isto é, que ocorre periodicamente somente para vizinhos diretos do nó emissor e com um intervalo adequado entre envios. No CDS-BitTorrent, um *peer* pode assumir dois papéis:

Peer Disseminador - responsável por escolher e disseminar algumas mensagens de PIECE via *broadcast* controlado, alcançando somente seus nós vizinhos na MANET.

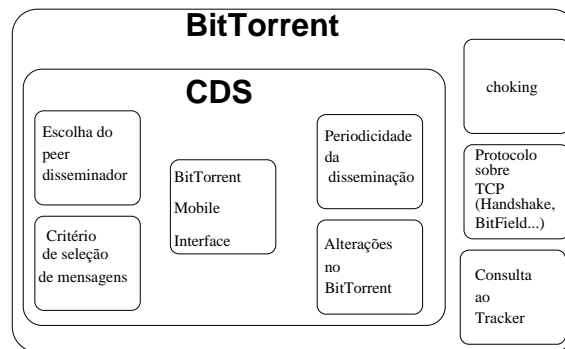


Figura 1. Arquitetura do CDS-BitTorrent

Como semente, o disseminador também poderá atender requisições por conteúdo de outros *peers* via *unicast*. É estabelecido que apenas uma das sementes deve ser o disseminador. O acréscimo de mais disseminadores do mesmo interesse poderia ser inconveniente, pois levaria a trocas extras de mensagens entre eles no intuito de evitar o envio de conteúdo redundante, sobrecarregando a rede.

Peer comum - não realiza disseminações via *broadcast* e pode receber mensagens de *PIECE*, sejam elas provenientes do disseminador, sejam resultantes de suas próprias requisições, mantendo sua autonomia na rede. Ele também pode se tornar semente e atender requisições por conteúdo de outros *peers* via *unicast*.

A Figura 1 mostra uma espécie de visão arquitetural do CDS-BitTorrent. O CDS-BitTorrent é composto por uma interface denominada BMI (*BitTorrent Mobile Interface*) e por elementos responsáveis por aplicar o critério de seleção de mensagens, escolher o *peer* disseminador e disseminar periodicamente mensagens selecionadas. Além disso, são necessárias algumas alterações no protocolo BitTorrent. Tudo isso é detalhado nas próximas seções.

4.1. Interface de Disseminação BMI

A BMI (*BitTorrent Mobile Interface*) é uma subcamada da camada aplicação. É responsável por lidar com mensagens de *PIECE* recebidas ou a serem enviadas via *broadcast*. As mensagens a serem enviadas pela BMI contam com o serviço não-orientado à conexão do UDP que, ao contrário do TCP, viabiliza a entrega de mensagens recebidas por *broadcast* à aplicação. A Figura 2 ilustra a comunicação entre as camadas aplicação, transporte e a interface BMI. Do lado emissor, a BMI recebe as mensagens a serem enviadas em modo *broadcast*. Essas mensagens são encapsuladas junto com um identificador de interesse do conteúdo (por exemplo, um valor de *hash* do arquivo compartilhado) em segmentos UDP. A BMI também sinaliza ao *socket* UDP o endereço de *broadcast* da rede como destino das mensagens. Do lado receptor, a BMI atua como um filtro, desencapsulando e repassando mensagens para a aplicação BitTorrent somente quando a mesma possuir interesse no conteúdo recebido. Caso não haja interesse, as mensagens correspondentes são descartadas pela BMI.

As mensagens destinadas ao envio em *unicast* são encaminhadas diretamente ao TCP sem passar pela BMI. Assim, um dispositivo utilizando o CDS-BitTorrent se serve tanto do TCP quanto da BMI/UDP para envio de mensagens. Os critérios de seleção de

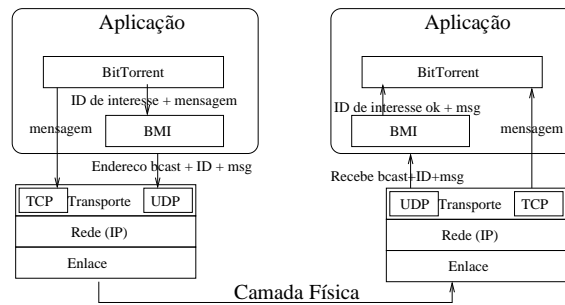


Figura 2. Pilha de protocolos com a interface BMI

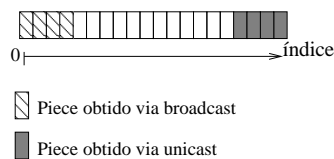


Figura 3. Peças com blocos baixados via *broadcast* e *unicast*

mensagens de PIECE a serem enviadas para a BMI é apresentado na próxima seção.

4.2. Critérios de Seleção de Mensagens

O CDS-BitTorrent dissemina parte das mensagens de PIECE em modo *broadcast*, uma vez que esse tipo de mensagem é o mais recorrente durante o *download*. Conforme observado na Seção 3.2, uma mensagem de PIECE carrega consigo um bloco de conteúdo. Assim, na prática, deve ocorrer uma seleção de quais blocos devem ser enviados em *broadcast* pelo disseminador e quais blocos devem ser requisitados por um *peer* comum. Tais seleções devem possuir regras distintas para evitar a chegada de mensagens redundantes que sobrecarregariam a rede. Assim, o CDS-BitTorrent estabelece as seguintes regras:

Seleção de blocos para disseminação - o disseminador escolhe preferencialmente os blocos mais raros dentre os *pieces* mais raros, do menor para o maior índice;

Seleção de blocos a serem requisitados - os *peers* comuns requisitam preferencialmente os blocos do menor para o maior *offset* dentre os *pieces* mais raros, do maior para o menor índice.

O disseminador armazena informações de blocos já disseminados e não envia um mesmo bloco mais de uma vez. Os *peers* conseguem estimar os blocos e *pieces* mais raros graças a uma modificação na mensagem de BITFIELD feita para o CDS-BitTorrent. Neste contexto, a mensagem de BITFIELD agora traz informação no nível de bloco. A mensagem de HAVE é utilizada para atualizar a informação de *pieces* baixados pelos demais *peers*. Com as estratégias adotadas, o preenchimento dos *pieces* em um *peer* próximo ao disseminador tende a ocorrer de acordo com a Figura 3. Contudo, a mobilidade pode potencialmente impactar essa tendência. Assim sendo, é necessária uma avaliação do desempenho do CDS-BitTorrent em cenários com mobilidade. Essa avaliação será apresentada na Seção 5.4.

4.3. Escolha do Peer Disseminador

A semente com maior identificador é escolhida como *peer* disseminador. Para o CDS-BitTorrent, o identificador do *peer* equivale ao seu endereço IP. Assim, quando uma se-

mente recebe a lista de *peers* vinda do *Tracker*, ela tem acesso aos seus identificadores. Para verificar se um *peer* é semente, são utilizadas as mensagens de BITFIELD, já presentes na implementação tradicional do BitTorrent. Tal critério confere simplicidade ao algoritmo por não exigir comunicação extra entre os *peers* para essa escolha. A cada mensagem BITFIELD recebida, a semente verifica se o emissor da mensagem também é uma semente. Se sim, compara os identificadores e, caso o seu seja menor, encerra o algoritmo. Caso contrário, se constatar que é a semente com maior identificador, assume o papel de disseminador. Uma vez que o CDS-BitTorrent é direcionado à situações de reuniões e conferências, é assumido que, inicialmente, todos os nós estão presentes. O procedimento é realizado apenas uma vez, nas primeiras trocas de BITFIELD, e é convencionalizado que o *Tracker* retorna todos os *peers* interessados no conteúdo. Ao contrário do que se pode imaginar, a existência de um único disseminador não afeta a tolerância a falhas do sistema, pois, mesmo que o disseminador deixe a rede ou que surjam partições na mesma, o *download* pode continuar através da comunicação *unicast*, quando possível.

4.4. Periodicidade de Disseminação

O envio em excesso de mensagens via *broadcast* pode gerar muitas colisões no meio de comunicação compartilhado, afetando negativamente o processo de *download*. Por isso, a periodicidade da disseminação por *broadcast* torna-se um fator fundamental para um bom desempenho do *download*. No CDS-BitTorrent, essa periodicidade é modelada como uma variável aleatória distribuída uniformemente no intervalo de tempo $[0, M]$, sendo M um parâmetro de projeto. A escolha do valor desse parâmetro será comentada na Seção 5.

4.5. Alterações no Protocolo BitTorrent

Conforme apresentado, o CDS-BitTorrent requer pequenas alterações no protocolo BitTorrent no tocante ao tratamento de mensagens BITFIELD e HAVE. Além disso, é necessário estender o BitTorrent com novas funcionalidades, como a escolha do *peer* disseminador e os algoritmos de seleção e disseminação de blocos.

5. Avaliações de Desempenho

Esta seção avalia, através de simulações com o NS-2 [Fall and Varadhan 2007], o desempenho do CDS-BitTorrent e do BitTorrent tradicional quando executados em um ambiente de MANETs. O módulo para NS-2 desenvolvido por Eger *et al.* [Eger *et al.* 2007] foi utilizado para a avaliação do BitTorrent tradicional¹. Para a avaliação do CDS-BitTorrent foram feitas diversas modificações nessa implementação, dentre elas, as apresentadas na Seção 4.5. Além delas, foi acrescido um módulo com protocolo de roteamento OLSR para as avaliações [Paquereau and Helvik 2006]. A modelagem das camadas física e MAC (*Medium Access Control*) segue a especificação IEEE 802.11g [IEEE Std 802.11g 2003]. O raio de alcance de comunicação dos nós é de 50 m para simular o alcance típico desse rádio em ambientes internos. A área onde os nós estão confinados é de $150\text{ m} \times 150\text{ m}$ para simular um espaço para eventos. O modelo de propagação de sinais utilizado é o *Two Ray Ground*. As métricas de avaliação, os cenários, os parâmetros de simulação e os resultados obtidos são apresentados nas próximas seções.

¹Nessa implementação, o *Tracker* é abstraído em um objeto visível a todos os nós da rede. Tal decisão não afeta os resultados deste trabalho, uma vez que o objetivo é a avaliação do desempenho do processo de *download* em si e não a do processo de localização de *peers*.

5.1. Métricas de Avaliação

As seguintes métricas foram utilizadas nas avaliações de desempenho:

Overhead de segmentos - número total de segmentos de transporte gerados até a entrega do conteúdo a todos os *peers*;

Fração de segmentos UDP e TCP - a fração de segmentos UDP e TCP gerados em comparação ao total de segmentos enviados com o uso do CDS-BitTorrent;

Pacotes perdidos - número de pacotes perdidos até a entrega total do conteúdo a todos os *peers*;

Tempo de download - tempo médio que um *peer* leva para finalizar o *download*;

Overhead de roteamento - número de mensagens de roteamento transmitidas, incluindo transmissões entre saltos;

Atraso médio fim a fim - tempo médio decorrido entre o envio e recebimento de segmentos de transporte até a completude do *download*, considerando o TCP para a abordagem tradicional e adicionalmente o UDP para o caso do CDS-BitTorrent.

5.2. Cenários e Demais Parâmetros de Simulação

Os cenários de simulação foram concebidos de forma a imitar uma situação de compartilhamento de um vídeo em um espaço para eventos, onde pessoas com dispositivos móveis formando uma MANET podem se mover ou permanecer paradas. Assim, dois cenários, um com mobilidade e outro sem, foram definidos para as avaliações de desempenho. Especificamente para o cenário sem mobilidade, os nós estão dispostos em uma topologia de grade, distribuídos uniformemente à distância de 1 m entre vizinhos de uma mesma fileira, imitando o espaçamento físico entre pessoas sentadas assistindo a uma palestra. Em específico para o cenário com mobilidade, 25 nós estão inicialmente dispostos em uma topologia de grade (5x5), distribuídos uniformemente à distância de 1 m entre vizinhos de uma mesma fileira.

Para simular a mobilidade, foi adotada uma variação do modelo *Random Waypoint* (RWP), conforme descrito a seguir: após um tempo de espera, os nós se movem a uma velocidade constante para um destino de coordenadas aleatórias para posteriormente voltarem a ficar imóveis durante o mesmo tempo de espera. Mais uma vez, após esse tempo de espera, os nós seguem para um novo destino aleatório, repetindo o processo descrito até o fim da simulação. O tempo de espera utilizado foi de 10 segundos. A diferença entre o modelo adotado e o RWP consiste no uso de uma velocidade constante, diferente deste último, onde, a cada mudança de direção, uma velocidade aleatória é escolhida em uma distribuição uniforme com zero como valor mínimo. Essa mudança foi feita motivada pelo fato de estudos mostrarem que a forma como o RWP tradicionalmente lida com a questão da velocidade é controversa [Yoon et al. 2003] e que a proximidade das velocidades a um valor estacionário em longas simulações levam a resultados mais realísticos, sendo este o caso do presente experimento.

Ambos os cenários avaliados possuem características em comum, as quais são descritas como segue. Cada nó executa uma instância da aplicação, ou seja, cada nó corresponde a um *peer* BitTorrent ou CDS-BitTorrent conforme abordagem avaliada. A simulação se inicia com apenas uma semente. Os nós entram na rede em instantes

aleatórios entre 0 e 1s após o início da simulação. Após a entrada de todos os nós na rede, não há saída ou entrada de novos nós. A simulação termina quando todos os nós concluem o *download*. O *peer* que finaliza um *download* continua na aplicação para colaborar com os demais. Durante a simulação, os *peers* compartilham um arquivo de 100MB, considerando-se *pieces* de 512KB, divididos em blocos de 16KB cada. No caso do CDS-BitTorrent, o parâmetro de periodicidade da disseminação M é igual a 1s. Tal valor foi obtido por meio de um estudo aqui omitido por limitações de espaço. No referido estudo, foi observado que um valor grande para M tornaria a disseminação pouco eficiente. Por outro lado, um valor de M menor que 1s desencadearia em uma sobrecarga de mensagens, degradando o desempenho na rede. O valor de $M = 1s$ garantiu em todos os cenários estudados um desempenho melhor do CDS-BitTorrent do que o do BitTorrent tradicional, sem sobrecarregar a rede. Os parâmetros específicos adotados para o BitTorrent foram os seguintes: intervalo de *choking* de 10 segundos e número máximo de *uploads* simultâneos igual a 4.

Todos os resultados apresentados a seguir possuem um intervalo de confiança de 99%. Os intervalos são representados por barras de erro nos gráficos, sendo algumas delas imperceptíveis. Cada ponto simulado corresponde à média obtida a partir de 20 simulações.

5.3. Cenário sem Mobilidade

No cenário sem mobilidade, as métricas de avaliação definidas foram estudadas em função do número de *peers* na rede. Foram realizadas simulações para 4 (2x2), 9 (3x3), 16 (4x4), 25 (5x5), 36 (6x6) e 49 (7x7) *peers* na rede.

A Figura 4(a) apresenta a fração de segmentos gerados pelo TCP e a partir de mensagens de *PIECE* enviadas à interface BMI durante o processo de *download* com o CDS-BitTorrent. Observa-se que apenas 5,4% do total de segmentos é enviado em *broadcast* no caso de haver 4 *peers* na rede. Esse valor diminui com o aumento do número de *peers*, atingindo 1,3% do total de segmentos para o caso de 49 *peers*. Essa diminuição ocorre porque com o aumento do número de *peers* há mais conexões TCP ativas, gerando mais tráfego em *unicast*, enquanto a geração de tráfego em *broadcast* pelo disseminador não varia em função do número de *peers* na rede.

A Figura 4(b) mostra o *overhead* de segmentos em função do número de *peers*. O CDS-BitTorrent apresenta um menor *overhead* de segmentos em relação à abordagem tradicional. Em particular, observa-se que ele gera 50% a menos de segmentos do que a abordagem tradicional com 36 *peers* na rede. Adicionalmente, observa-se que a taxa de crescimento de *overhead* de segmentos do CDS-BitTorrent também é menor do que a da abordagem tradicional. O melhor desempenho do CDS-BitTorrent se justifica pelo fato do pequeno percentual de segmentos enviados em *broadcast* servir todos os *peers* da rede e assim permitir uma redução na quantidade de requisições por blocos e no consequente tráfego em *unicast* na rede.

A Figura 4(c) mostra como o número de pacotes perdidos na rede varia em função do número de *peers*. Nota-se que a perda de pacotes com o CDS-BitTorrent é até 40% menor do que a perda observada com a abordagem tradicional. Isso é consequência direta da menor quantidade de segmentos que o mesmo gera e envia na rede.

A Figura 4(d) apresenta o tempo de médio *download* em função do número de

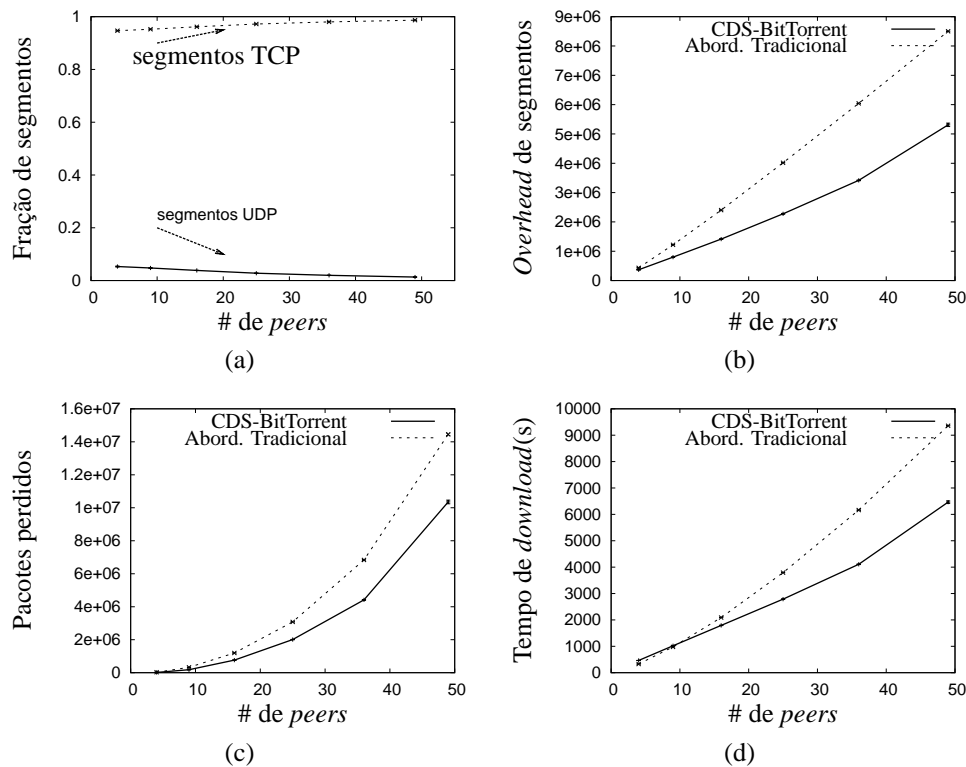


Figura 4. Cenário sem Mobilidade: (a) Fração de segmentos (b) Overhead de segmentos (c) Pacotes perdidos (d) Tempo de download

peers. Nota-se que a taxa de crescimento do tempo de *download* é menor com o uso do CDS-BitTorrent. Observa-se ainda que não há diferença significativa entre as abordagens até 9 *peers* na rede. Entretanto, a partir de 16 *peers*, o tempo de *download* passa a ser menor para o CDS-BitTorrent. No caso de 49 *peers*, por exemplo, observa-se uma melhora de 30% em comparação à abordagem tradicional. O melhor tempo de *download* observado com o uso do CDS-BitTorrent é consequência direta do menor *overhead* de segmentos que ele gera para a entrega do conteúdo.

A Figura 5(a) apresenta o *overhead* de roteamento em função do número de *peers*. Observa-se que o uso do CDS-BitTorrent contribui para a diminuição do *overhead* de roteamento. Em particular, esse *overhead* chega a ser 20% menor para o caso de 49 *peers*. O menor *overhead* em comparação à abordagem tradicional é consequência direta do menor tempo de *download* proporcionado pelo uso do CDS-BitTorrent já que, nesse período de tempo mais curto, haverá menos pacotes de roteamento gerados.

A Figura 5(b) mostra o atraso médio fim a fim em função do número de *peers*. Observa-se que tal atraso é similar em ambas as abordagens para até 36 *peers*. A partir desse valor, o atraso médio fim a fim é ligeiramente maior com o CDS-BitTorrent, sendo enfatizado pela nítida mudança na inclinação da curva. Note que isso se reflete também na mudança de inclinação que ocorre no mesmo ponto da curva do tempo de *download* apresentada na Figura 4(d). O aumento no atraso médio fim a fim a partir de 36 *peers* se justifica pelo aumento de colisões de transmissões por causa do *broadcast* já que a disputa pelo meio de comunicação se torna mais intensa à medida que o número de *peers* aumenta. Apesar disso, o ligeiro aumento observado no atraso médio fim a fim não tem

impacto significativo no tempo médio de *download* com o CDS-BitTorrent já que com o mesmo, a quantidade de segmentos gerados é bem menor.

5.4. Cenário com Mobilidade

No cenário com mobilidade, as métricas de avaliação definidas foram estudadas em função da velocidade dos nós da rede. Foram realizadas simulações para as velocidades de $0,0\text{ m/s}$ a $1,5\text{ m/s}$ em passos de $0,5\text{ m/s}$, considerando 25 peers .

A Figura 6(a) mostra como a fração de segmentos gerados a partir de mensagens de *PIECE* enviadas à interface *BMI* varia em função da velocidade do nós durante o *download* com o CDS-BitTorrent. Observa-se que, em média, de 2,8% (para $0,0\text{ m/s}$) a 2,0% (para $1,5\text{ m/s}$) do total de segmentos gerados são originados a partir de conteúdo proveniente da interface *BMI*. A diminuição do percentual ocorre por causa do aumento do volume de tráfego TCP, já que o volume de tráfego enviado em *broadcast* pelo disseminador não depende da velocidade dos nós. Como a mobilidade espalha os nós pela área de $150\text{m} \times 150\text{m}$, diminuindo o número de nós vizinhos do nó disseminador, menos *peers* são favorecidos continuamente pela disseminação. Isso obriga os demais *peers* a fazerem mais requisições por conteúdo, gerando um maior volume de tráfego TCP na rede em resposta a tais requisições.

A Figura 6(b) mostra o *overhead* de segmentos em função da velocidade dos nós. Observa-se que esse *overhead* é significativamente menor com o uso do CDS-BitTorrent. Porém, o *overhead* de segmentos é aproximadamente o mesmo com o aumento da velocidade para o caso da abordagem tradicional ao passo que para o caso de uso do CDS-BitTorrent, há um aumento quando há mobilidade. Isso ocorre com o uso do CDS-BitTorrent por causa do aumento da fração de tráfego TCP na rede como explicado anteriormente. Apesar disso, o *overhead* de segmentos com o CDS-BitTorrent no pior dos casos foi 20% menor do que o alcançado com a abordagem tradicional.

A Figura 6(c) mostra o número de pacotes perdidos em função da velocidade. Observa-se que há uma menor quantidade de pacotes perdidos com uso do CDS-BitTorrent para todas as velocidades estudadas. Observa-se também que para ambas as abordagens, o número de pacotes perdidos diminui com o aumento da velocidade. A explicação é como segue: o aumento da velocidade faz com que os *peers* deixem de compartilhar mais rapidamente o mesmo domínio inicial de colisões. Em consequência, a

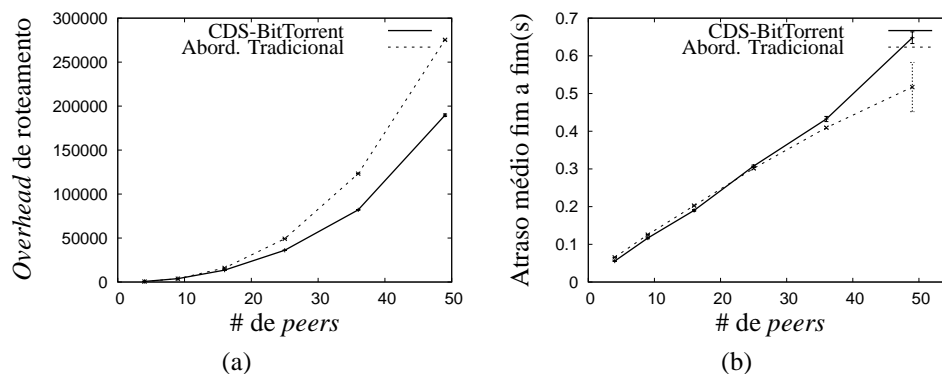


Figura 5. Cenário sem Mobilidade: (a) *Overhead* de roteamento (b) Atraso médio fim a fim

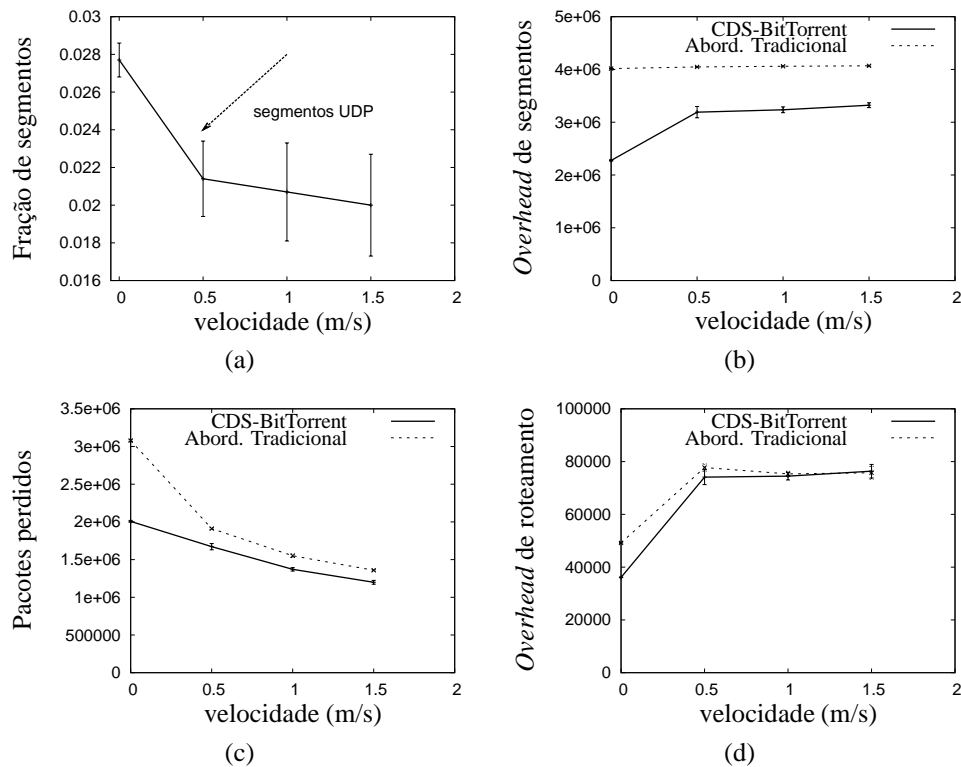


Figura 6. Cenário com Mobilidade: (a) Fração de segmentos (b) Overhead de segmentos (c) Pacotes perdidos (d) Overhead de roteamento

disputa inicial pelo meio de comunicação sem fio é amenizada mais rapidamente, reduzindo perdas por colisão de transmissões.

A Figura 6(d) mostra o *overhead* de roteamento em função da velocidade. Nota-se que para ambas as abordagens há um aumento abrupto do *overhead* de roteamento com a mobilidade, passando o mesmo a ser similar nos dois casos a partir da velocidade de $0,5 \text{ m/s}$. O aumento abrupto é consequência da reação do OLSR à mudanças de topologia da rede por causa da mobilidade, o que o faz reduzir o intervalo de envio periódico de mensagens de controle na rede. Já a similaridade do *overhead* a partir de $0,5 \text{ m/s}$, se justifica pelo tempo de *download* similar (Figura 7(a)) entre ambas as abordagens, tornando o tempo de execução do protocolo de roteamento para envio de mensagens periódicas também similar durante as simulações.

A Figura 7(a) mostra o tempo médio de *download* em função da velocidade. Observa-se que este tempo não varia significativamente com a velocidade dos nós quando a abordagem tradicional é utilizada. Por outro lado, no caso do CDS-BitTorrent, o tempo de *download* apresenta um aumento quando se adiciona mobilidade ao cenário. O aumento é consequência do espalhamento dos nós que, por sua vez, diminui o número de *peers* continuamente servidos pelo conteúdo enviado em *broadcast* e faz com que o tráfego *unicast* tenha participação maior no processo de obtenção do arquivo, aumentando o tempo de *download*. Vale notar que o tempo de *download* com o CDS-BitTorrent é similar ao da abordagem tradicional para velocidades de $0,5 \text{ m/s}$ a $1,5 \text{ m/s}$. A princípio, esse resultado não seria esperado pois, conforme apresentado, o *overhead* de segmentos e a perda de pacotes são menores com o uso do CDS-BitTorrent para tais velocidades. A

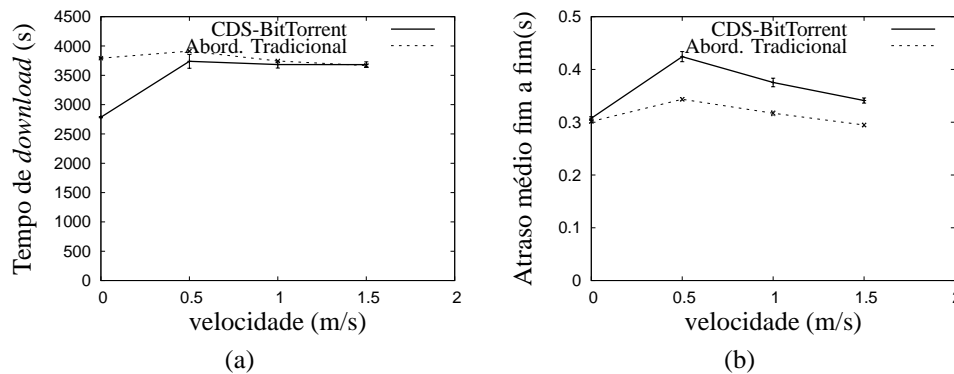


Figura 7. Cenário com Mobilidade: (a) Tempo de download (b) Atraso médio fim a fim

razão para o ocorrido se deve ao fato do tempo médio de *download* passar a ser dominado pelos atrasos na entrega de pacotes em consequência de atrasos para a reconstrução de rotas e do aumento abrupto do *overhead* de roteamento com a mobilidade, gerando mais tráfego na rede. Apesar do exposto, o tempo médio de *download* com o CDS-BitTorrent é sempre menor ou igual ao obtido com a abordagem tradicional.

A Figura 7(b) mostra o atraso médio fim a fim de segmentos em função da velocidade. Observa-se que tal atraso aumenta para ambas as abordagens avaliadas quando se passa de $0,0\text{ m/s}$ a $0,5\text{ m/s}$. Isto ocorre pois passam a existir atrasos para a reconstrução de rotas e atrasos consequentes do tráfego adicional na rede gerado pelo crescimento abrupto do *overhead* de roteamento. Observa-se também uma ligeira queda no atraso médio fim a fim para ambas as abordagens a partir de $0,5\text{ m/s}$. Essa queda é decorrente da redução da perda de pacotes observada com o aumento da mobilidade conforme Figura 6(c). Nota-se ainda que o impacto da mobilidade no atraso médio fim a fim é maior com o uso do CDS-BitTorrent. O maior impacto é consequência direta do fato do aumento relativo do *overhead* de roteamento ser maior com o uso do CDS-BitTorrent do que com o uso da abordagem tradicional ao se passar de um cenário sem mobilidade para um com mobilidade conforme mostra a Figura 6(d). Em suma, os resultados mostram que o atraso médio fim a fim é maior com o uso do CDS-BitTorrent no cenário estudado. Entretanto, isso não trouxe impacto suficiente no tempo de *download* para que o mesmo fosse maior do que o obtido utilizando-se a abordagem tradicional já que houve uma compensação pelo fato do CDS-BitTorrent gerar um menor *overhead* de segmentos.

6. Conclusões

Este trabalho introduziu o sistema CDS-BitTorrent que se propõe à melhoria do processo de *download* do BitTorrent em MANETs através da adoção de estratégias específicas de seleção e disseminação de conteúdo. A redução no tempo de *download* com o uso do CDS-BitTorrent se mostrou mais efetiva nos cenários sem mobilidade e de baixa mobilidade (até $0,5\text{ m/s}$), sendo adequado aos cenários para os quais foi proposto. Ainda sim, é importante ressaltar que o CDS-BitTorrent obteve em todos os cenários avaliados, independente da mobilidade, um menor *overhead* de segmentos e uma menor perda de pacotes na rede, contribuindo diretamente para um menor consumo de energia dos nós da MANET. De forma geral, os resultados mostraram que disseminar seletivamente apenas

de 1,3% a 5,4% do volume total de segmentos via um *broadcast* “racional”, traz impactos positivos de desempenho, enfatizando a boa relação custo/benefício das estratégias adotadas pelo CDS-BitTorrent. O CDS-BitTorrent ainda mantém a tolerância a falhas do sistema, ao permitir que *peers* atuem de forma independente da disseminação.

O BitTorrent foi concebido para a Internet e considera que todas as conexões possuem um mesmo custo. Ao se valer de difusões via UDP, o CDS-BitTorrent diminui a importância de conexões TCP para diversos *peers* e consegue assim melhorar o desempenho. Os resultados aqui apresentados abrem caminho para discussões sobre até que ponto os princípios do BitTorrent para Internet são adequados em MANETs. Vale ressaltar que o CDS-BitTorrent é um sistema em evolução, mas que as ideias aqui apresentadas são promissoras conforme sugerem os resultados apresentados. Em trabalhos futuros, extensões ao CDS-BitTorrent podem ser consideradas, como o ajuste automático do parâmetro M em função do tráfego na rede e o controle da disseminação baseado em Teoria dos Jogos. Além disso, pode-se ampliar a variedade de cenários de estudo, incluindo situações com diferentes velocidades entre os nós e diferentes interesses por conteúdo.

Referências

- Cohen, B. (2006). BitTorrent Specification. Technical report, BitTorrent.org.
- Eger, K., Hossfeld, T., Binzenhofer, A., and Kunzmann, G. (2007). Efficient Simulation of Large-scale P2P Networks: Packet-level vs. Flow-level Simulations. In *Proceedings of the 2nd UPGRADE-CN*, pages 9–16, New York, NY, USA.
- Fall, K. and Varadhan, K. (2007). NS Notes and Documentation. Technical report, The VINT Group.
- Hu, Y. C., Das, S. M., and Pucha, H. (2005). Peer-to-Peer Overlay Abstractions in MANETs. In Wu, J., editor, *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, volume 1, pages 857–874. CRC Press.
- IEEE Std 802.11g (2003). IEEE Standard.
- Krifa, A., Sbai, M. K., Barakat, C., and Turletti, T. (2009). BitHoc: A Content Sharing Application for Wireless Ad hoc Networks. In *Proceedings of the IEEE Percom*.
- Paquereau, L. and Helvik, B. E. (2006). A Module-based Wireless Node for NS-2. In *Proceedings of the WNS2*, page 4, New York, NY, USA.
- Rajagopalan, S. and Shen, C.-C. (2006). A Cross-layer Decentralized BitTorrent for Mobile Ad hoc Networks. In *Proceedings of the 3rd MobiQuitous*, pages 1–10.
- Souza, C. and Nogueira, J. M. (2008). Um Estudo do BitTorrent em Redes ad hoc sem Fio Críticas com Localidade Espaço-temporal. In *Anais do 25º SBRC*, pages 329–342.
- Yoon, J., Liu, M., and Noble, B. (2003). Random Waypoint Considered Harmful. In *Proceedings of the IEEE INFOCOM*, volume 2, pages 1312–1321.