

Um estudo do BitTorrent em redes ad hoc sem fio críticas com localidade espaço-temporal*

Carlos Souza, José Marcos Nogueira

¹Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) - Belo Horizonte - Brasil

{chu, jmarcos}@dcc.ufmg.br

Abstract. *Critical networks are ad hoc networks established autonomously to provide communication in emergencies. In these situations, where the devices have low resources and the network is failure-prone, P2P networks are the best architecture to data dissemination since they are resilient and efficient. In this paper, we study this scenario and show that the BitTorrent network has properties that make it an ideal candidate to be used in critical networks. Meanwhile, BitTorrent can't deal with spatial-temporal locality, a typical property of file transfer applications in critical networks, consisting of many nodes close to each other downloading the same file at the same time. We propose an extension of BitTorrent that solves this problem and improves the BitTorrent performance.*

Resumo. *Redes críticas são redes ad hoc formadas de forma autônoma para prover comunicação em situações de emergência. Nessas situações, onde os dispositivos são limitados computacionalmente e a rede é propensa a falhas, redes P2P são a melhor arquitetura para disseminação de dados pois são resilientes e eficientes. Neste trabalho estudamos este cenário e mostramos que o BitTorrent possui propriedades que fazem dele um candidato ideal para ser empregado em redes críticas. Entretanto, o BitTorrent não é capaz de lidar com localidade espaço-temporal, uma propriedade presente em aplicações de transferência de arquivos em redes críticas, que consiste em vários nós próximos descarregando o mesmo arquivo ao mesmo tempo. Propomos uma modificação do BitTorrent para resolver este problema e melhorar a sua eficiência nestas situações.*

1. Introdução

Redes críticas são redes ad hoc formadas de forma autônoma e rápida para prover infraestrutura de comunicação em situações de emergência onde não existe infraestrutura prévia ou cuja infraestrutura foi danificada. Exemplos de aplicações dessas redes são a coordenação de missões de resgate e de resposta a desastres naturais, segurança pública, defesa civil e a presença de equipes de pesquisa em regiões isoladas. Ao utilizar uma infraestrutura de comunicação, é possível obter uma maior coordenação e integração das equipes, evitando assim repetição de esforços e diminuindo o tempo de tomada de decisões. A rede pode transportar dados que vão auxiliar o planejamento das equipes, como medidas coletadas por redes de sensores sem fio. Os usuários da rede, por exemplo,

*Projeto parcialmente suportado pela FAPEMIG - bolsa de IC e projeto de pesquisa Radhocri (Redes ad hoc para missões críticas)

bombeiros ou policiais, possuem dispositivos móveis pessoais (notebooks e PDAs) os quais são utilizados para fins de comunicação e acesso aos dados.

Uma rede crítica deve utilizar os seus recursos de forma eficiente pois por ser uma rede de implantação rápida supõe-se que possui recursos limitados. Deve ser auto-configurável, auto-organizável e auto-gerenciada pois os usuários não possuem conhecimentos técnicos em informática e ela não deve exigir o emprego de pessoas especializadas na manutenção da rede, pois todo o pessoal deve trabalhar para resolver os problemas da situação de emergência. Ela ainda deve ser robusta pois será empregada em situações onde falhas de comunicação podem levar a um agravamento da situação.

Em algumas situações os participantes da rede precisam descarregar (realizar o *download*) arquivos, por exemplo mapas gerados por redes de sensores, para auxiliar a execução de suas tarefas. Aplicações de transferência de arquivos em redes críticas apresentam *localidade espaço-temporal*, ou seja, nós próximos tendem a realizar transferências dos mesmos arquivos ao mesmo tempo. Isso ocorre pois a rede é formada por usuários reunidos em equipes (médicos, policiais ou bombeiros por exemplo) com propósitos comuns e que estão próximos e cooperam entre si para realizar a mesma tarefa. Quando um usuário descarrega um arquivo para realizar uma tarefa, espera-se que os outros usuários da equipe queiram descarregar o mesmo arquivo pois estão envolvidos na mesma tarefa. Por exemplo, em uma equipe de bombeiros que se prepara para entrar em um prédio em chamas, se um bombeiro descarrega a planta do prédio para seu PDA, é de se esperar que os outros bombeiros também descarreguem o mesmo arquivo.

Em redes críticas, é desaconselhável utilizar uma arquitetura cliente-servidor para transferências de arquivos. Essa arquitetura cria um ponto único de vulnerabilidade e um gargalo no servidor, exige o uso de dispositivos com recursos computacionais abundantes para realizar o papel de servidor e não funciona no caso de particionamento da rede. O uso de uma rede par a par (peer to peer - P2P) para compartilhamento de arquivos distribui a tarefa de servir arquivos entre os participantes da rede P2P resolvendo o problema do ponto único de falha e do gargalo, elimina a necessidade de haver um servidor dedicado e torna a rede mais resistente a particionamentos ao permitir que nós descarreguem um arquivo a partir de qualquer outro nó que possua o arquivo ou parte dele.

Acreditamos que o BitTorrent é uma rede de compartilhamento de arquivos ideal para redes críticas. Ele sobrecarrega pouco a rede com mensagens de controle e se preocupa em melhorar o desempenho das transferências de dados. Mostramos como estas características se relacionam com redes críticas na seção 3.2. Contudo, o BitTorrent não está preparado para lidar com localidade espaço-temporal em redes ad hoc sem fio. Como há muitos nós próximos realizando transferências de arquivos, o meio torna-se congestionado, o que causa redução das taxas de transferência. Como a política de incentivos do BitTorrent leva em conta as taxas de transferência dos participantes, esses nós encontram dificuldades em interagir com os demais participantes. Além disso, o BitTorrent não é capaz de aproveitar relações com os nós próximos para melhorar o desempenho da rede.

Neste trabalho propomos uma modificação do BitTorrent para melhorar o seu desempenho em redes ad hoc sem fio com localidade espaço-temporal. Esta modificação consiste em agrupar nós vizinhos descarregando o mesmo arquivo e coordenar a transferência do arquivo entre eles. Trata-se de uma abordagem *cross-layer*, ou seja, exploramos

recursos de várias camadas da pilha de protocolos.

O restante deste trabalho está organizado da seguinte forma: revisamos os principais trabalhos relacionados na seção 2 e descrevemos o BitTorrent e a sua utilização em redes críticas na Seção 3. Na Seção 4 mostramos os problemas encontrados pelo BitTorrent diante de localidade espaço-temporal e descrevemos uma extensão do BitTorrent para solucionar estes problemas na Seção 5. Por fim avaliamos esta solução na Seção 6.

2. Revisão da literatura

A relação entre redes P2P e redes ad hoc sem fio é um tema bem explorado na literatura. [Y. Charlie Hu and Pucha 2003] apontam uma sinergia entre essas redes. Ambas são descentralizadas e auto-organizáveis com topologia dinâmica e responsáveis por realizar roteamento em um ambiente distribuído. Mais do que isso, [Oliveira et al. 2005] argumentam que estas redes são complementares. Redes ad hoc são geralmente compostas de dispositivos com recursos limitados enquanto redes P2P eliminam a necessidade de servidores com recursos abundantes. Além disso, redes ad hoc sem fio são suscetíveis a falhas enquanto redes P2P são tolerantes a falhas da rede subjacente.

[da Hora et al. 2007] apontam o uso de redes P2P sobre redes ad hoc sem fio como o mecanismo ideal de disseminação de informação em situações de emergência. Devido às características desses cenários, o uso de uma arquitetura cliente-servidor é inviável pois o servidor se torna um ponto único de vulnerabilidade. Além disso, os nós de uma rede ad hoc sem fio possuem baixo poder computacional e portanto são incapazes de atuar como servidores por muito tempo ou de atender várias requisições simultaneamente. O uso de uma arquitetura P2P distribui a tarefa de servir o arquivo entre os nós da rede reduzindo a possibilidade de sobrecargas e eliminando o ponto único de vulnerabilidade.

Há várias propostas de redes P2P para redes ad hoc sem fio. No melhor do nosso conhecimento, não existe nenhum trabalho na literatura que aborde a questão da localidade espaço-temporal. Porém há vários trabalhos que propõem abordagens cross-layer para este problema. Em [Conti et al. 2005], é proposta uma otimização *cross-layer* do Gnutella em que os nós escolhem como seus vizinhos lógicos os vizinhos da rede subjacente. Os resultados desse trabalho mostram que essa abordagem reduz o tempo de pesquisa e aumenta a escalabilidade da rede.

Em [Delmastro 2005] é proposta uma modificação *cross-layer* da rede P2P Pastry a fim de aumentar sua eficiência em redes ad hoc. Essa modificação consiste em acoplar a rede P2P com a rede subjacente. Cada nó Pastry conhece todos os outros nós da rede graças ao uso de um protocolo de roteamento pró-ativo. As mensagens de controle da rede P2P são transportadas junto com as mensagens de controle desse protocolo facilitando a tarefa de manter a rede sobreposta consistente e diminuindo o tráfego na rede.

[Rajagopalan and Shen 2006] argumentam que o BitTorrent possui várias características que o tornam um bom candidato a ser utilizado em redes ad hoc sem fio. Possui uma arquitetura que amortiza o custo de servir o arquivo por toda a rede, permite reiniciar transferências após desconexões da rede e possibilita que um nó sirva um arquivo enquanto ele ainda o está descarregando. Porém o BitTorrent ainda possui estruturas centralizadas que são pontos únicos de vulnerabilidade, como o *tracker* e o servidor de buscas. Os autores apresentam uma modificação *cross-layer* do BitTorrent em que a pesquisa por

arquivos e por outros nós é realizada por inundação, eliminando assim qualquer estrutura centralizada. Para aumentar a eficiência das pesquisas e aumentar a disponibilidade dos dados, os arquivos são replicados em outros nós.

3. BitTorrent

O BitTorrent [Cohen 2003] é uma rede P2P de compartilhamento de arquivos com foco no desempenho na transferência de dados e não na localização de conteúdo, como o Gnutella e Pastry. Descrevemos o seu funcionamento na Seção 3.1 e mostramos como ele se presta a resolver o problema de transferências de arquivos em redes críticas na seção 3.2.

3.1. Funcionamento

No BitTorrent, os arquivos são divididos em pedaços chamados *pieces*. Cada participante da rede BitTorrent descarrega os *pieces* de participantes que ainda estão descarregando o arquivo, os *leechers*, e de participantes que possuem o arquivo completo, as *sementes*. Assim o custo de servir um arquivo é amortizado entre os participantes da rede.

Quando um participante inicia a transferência de um arquivo, ele informa a um participante especial da rede, o *tracker*, o seu endereço e obtém o endereço de outros *leechers* e de sementes. Para descarregar um arquivo, o usuário deve possuir o seu *torrent*, que é um arquivo que contém uma descrição do arquivo, seu tamanho, tamanho dos *pieces*, o identificador do arquivo e o endereço do *tracker*. O BitTorrent não especifica como é obtido o *torrent*. Ele pode ser obtido através de um site por exemplo.

O BitTorrent utiliza uma política de incentivos para fazer com que os participantes contribuam com o resto da rede. Cada participante envia dados para os participantes que mais colaboraram com ele. Ele escolhe aqueles com as maiores taxas de envio. Uma semente utiliza como critério a taxa de recepção dos participantes. Dessa forma, ela consegue disseminar mais rapidamente um arquivo pela rede.

3.2. BitTorrent e redes críticas

O BitTorrent possui propriedades que fazem dele uma boa rede P2P de distribuição de dados em redes críticas. Ele é mais resistente a falhas de conectividade. Como o arquivo é dividido em pedaços, caso um nó perca comunicação temporariamente, é possível recomençar a transferência mais tarde, bastando descarregar os pedaços que faltam. Isto é importante em redes ad hoc sem fio, que são propensas a falhas devido à má qualidade do meio e à mobilidade dos nós.

O BitTorrent sobrecarrega pouco a rede com mensagens de controle. Redes como Pastry e CAN precisam trocar mensagens constantemente para manter a estrutura complexa da rede sobreposta. Quando há entrada e saída frequente de nós, como é o caso de redes ad hoc sem fio, o volume de mensagens necessárias para manter a estrutura da rede sobreposta consistente representa um custo bastante alto, o que pode inviabilizar a utilização dessas redes P2P. Os participantes da rede BitTorrent, por outro lado, precisam trocar mensagens apenas entre alguns poucos nós e somente enquanto há alguma transferência em progresso.

Um inconveniente da sua utilização é a presença de estruturas centralizadas como o *tracker* e o servidor de pesquisas, que são pontos únicos de vulnerabilidade. Para resolver este problema, utilizamos a abordagem proposta em [Rajagopalan and Shen 2006].

A pesquisa centralizada por *torrents* e demais participantes é substituída por pesquisas através de inundação eliminando os pontos únicos de vulnerabilidade. Para aumentar a eficiência das pesquisas e a disponibilidade dos dados, os arquivos de conteúdo são replicados em outros nós.

4. BitTorrent em presença de localidade espaço-temporal

Nesta seção são descritos os problemas encontrados pelo BitTorrent em redes ad hoc sem fio diante de localidade espaço-temporal. Esses problemas fazem com que ele tenha um desempenho inferior em redes críticas e utilize os recursos da rede de forma dispendiosa.

4.1. Redução da competitividade por redução das taxas de transferência

Uma vez que há nós próximos realizando transferências de arquivo ao mesmo tempo, o meio torna-se congestionado e a largura de banda obtida por cada nó é reduzida. Como os participantes do BitTorrent privilegiam os nós com maior taxa de envio e, no caso das sementes, os nós com a maior taxa de recepção, estes nós encontram dificuldades para descarregar arquivos da rede BitTorrent. Isso faz com que estes nós demorem mais tempo para realizar transferências do que nós que estão em uma região menos congestionada.

4.2. Interação reduzida entre nós próximos

Como há localidade espaço-temporal em transferências de arquivo em redes críticas, a probabilidade de nós fisicamente próximos estarem descarregando o mesmo arquivo é alta. Neste caso é importante que os nós dêem preferência a conexões com nós próximos em relação a conexões com nós distantes para descarregar o arquivo. Assim, a rede fica menos sobrecarregada e o atraso das transferências torna-se menor. Além disso, a probabilidade de haver uma ruptura na comunicação entre nós vizinhos é menor do que entre nós distantes pois há menos saltos até o destino neste caso. O BitTorrent porém não possui mecanismos para privilegiar transferências entre nós próximos. Na verdade, o BitTorrent não é capaz de distinguir nós distantes de nós próximos pois está situado na camada de aplicação e não consegue enxergar a rede subjacente. Daí a necessidade de utilizar uma abordagem *cross-layer*.

Além disso, na implementação padrão do BitTorrent, cada nó se comunica apenas com um conjunto aleatório de nós informado pelo *tracker*. Não há garantias de que os nós vizinhos estejam neste conjunto. Assim um nó pode não se comunicar com um vizinho simplesmente por não saber de sua existência. Este problema não existirá mais caso a implementação proposta por [Rajagopalan and Shen 2006] seja utilizada, onde o *tracker* é eliminado e a pesquisa por outros nós é feita por inundação.

4.3. Transmissões redundantes

Quando há nós próximos entre si, uma mensagem contendo uma parte do arquivo que é recebida por um nó também alcança os nós vizinhos, considerando a comunicação sem fio na modalidade difusão. Logo uma única mensagem seria suficiente para enviar este pedaço do arquivo a todos os nós desta vizinhança que estiverem descarregando um mesmo arquivo. Porém, como as transferências são feitas de forma independente, o arquivo é transmitido várias vezes localmente de forma redundante.

Seria mais eficiente enviar o arquivo utilizando mensagens em modo *multicast*. O BitTorrent porém não é capaz de enviar este tipo de mensagem pois está situado na

camada de aplicação, enquanto este é um serviço da camada de acesso ao meio. Logo deve ser adotada uma abordagem *cross-layer*.

5. Proposta de extensão do BitTorrent

Para solucionar os problemas apresentados pelo BitTorrent diante de localidade espaço-temporal em redes ad hoc sem fio, propomos uma extensão do BitTorrent. Esta solução consiste em estender o protocolo para que seja criado um grupo com nós vizinhos para cada transferência de arquivos e, dentre estes nós, eleger um líder que é responsável por descarregar o arquivo da rede P2P e distribuir o arquivo para o grupo utilizando transmissão em modo *multicast*. Portanto, todos os nós do grupo devem ser alcançáveis pelo líder do grupo com mensagens em modo *multicast*. Essa solução exige a modificação do BitTorrent apenas nos nós que farão parte do grupo, que são capazes de trocar arquivos com nós convencionais, inclusive fora da rede crítica.

Utilizamos uma abordagem *cross-layer*, ou seja, exploramos recursos de várias camadas da pilha de protocolos. Da camada de rede, utilizamos a informação sobre a topologia da rede para que o cliente BitTorrent saiba quem são os seus vizinhos que estão descarregando o mesmo arquivo. Da camada de acesso ao meio, utilizamos o modo *multicast* para transmitir o arquivo para todos os nós do grupo com uma única mensagem.

Os outros nós do grupo param de se comunicar com o resto da rede P2P, congestionando menos o meio e permitindo que o nó líder consiga taxas de envio e recepção maiores. Assim, ele descarrega o arquivo mais facilmente de outros participantes da rede e das sementes. Essa solução explora bem as relações entre nós vizinhos. Todos os nós do grupo só estão conectados ao líder e o líder só se comunica com participantes fora do grupo para descarregar *pieces* que nenhum nó do grupo possui. Uma vez que as mensagens são transferidas em modo *multicast* para todos os nós do grupo, cada pedaço do arquivo é transmitido uma única vez e não há mais transmissões redundantes.

O restante desta seção detalha o processo de criação do grupo e de transferência de arquivos.

5.1. Criação do grupo

O processo de criação do grupo consiste em fazer com que todos os nós numa determinada vizinhança física descubram quem são os seus vizinhos imediatos, ou seja, aqueles nós com os quais ele consegue se comunicar sem a ajuda de nós intermediários, que estão descarregando o mesmo arquivo. Então os nós dessa vizinhança decidem através de um processo de eleição quem será o líder.

5.1.1. Encontrando os vizinhos

Quando um nó inicia a transferência de um arquivo, ele tenta descobrir periodicamente os seus vizinhos imediatos que estão descarregando o mesmo arquivo. Isto é feito enviando uma mensagem em modo *broadcast*, ENCONTRA_VIZINHOS, com o identificador do arquivo. Um nó que esteja descarregando este arquivo responde através de uma mensagem VIZINHO. Então este nó tenta encontrar os seus próprios vizinhos, mas apenas se ele não tiver feito isto há pelo menos t_v segundos. Esta limitação impede que este processo se repita indefinidamente.

Caso um nó que já esteja trabalhando como o líder de um grupo receba a mensagem ENCONTRA_VIZINHOS com o identificador do arquivo que ele está distribuindo, ele responde com a mensagem GRUPO convidando este nó para se juntar ao seu grupo.

5.1.2. Eleição do líder

Para selecionar o líder do grupo, utilizamos um algoritmo de eleição simples. Após enviar uma mensagem ENCONTRA_VIZINHOS, cada nó espera t_g segundos e, se receber a resposta de pelo menos dois vizinhos, ele inicia a fase de eleição concorrendo para ser o líder do grupo. Baseado na resposta dos vizinhos, cada candidato computa o valor, u , de uma função utilidade que determina quão bom ele seria para a rede se fosse o líder do grupo. A função utilidade será discutida na seção 5.1.3. Ele envia o valor de u para os seus vizinhos e recebe deles os seus valores de u .

Os nós elegem independentemente seus líderes, que são aqueles que enviaram o maior valor de u . No caso de empate, o nó com o menor identificador é eleito líder. Então os nós enviam uma mensagem, SUBMISSAO, ao líder escolhido. Após enviar a sua função utilidade, cada candidato espera t_e segundos por mensagens de submissão. Um nó que receber mensagens de submissão de pelo menos dois nós se declara líder enviando para estes nós a mensagem LIDER. Caso um nó seja eleito líder, mas não aceite esta tarefa, por exemplo porque ele mesmo elegeu um outro nó como líder, ele envia uma mensagem REJEICAO para todos os nós que o elegeram. Os nós rejeitados reiniciam então o processo de criação do grupo.

O líder desempenha um papel mais ativo que os outros nós do grupo e por isso consome mais recursos que esses. Isto é uma questão importante principalmente considerando o consumo de energia, que é crítico em redes sem fio. Pode-se argumentar que não há incentivos para que um nó assuma o papel de líder, porém, uma vez que todos os dispositivos são propriedade da mesma organização, eles podem ser programados para trabalhar em conjunto. Logo não há problema em assumir que um nó vai se comportar de maneira altruísta.

Embora um esquema de revezamento de líderes possa distribuir o consumo de energia entre os nós do grupo, isso pode reduzir a eficiência da solução. O líder carrega consigo um histórico junto ao resto da rede P2P. Ele enviou *pieces* para outros participantes da rede P2P e assim também recebe *pieces* destes participantes. Quando ocorre troca de líder, o novo líder atua como se fosse um novo nó na rede P2P e portanto vai demorar para conseguir descarregar o arquivo, considerando a política de incentivos do BitTorrent. Além disso, os outros nós podem não ter a mesma visão do grupo que o líder, ou seja, podem não alcançar através de uma mensagem em modo *multicast* todos os nós do grupo, e neste caso são incapazes de atuar como líderes.

5.1.3. Função utilidade

Há vários fatores que podem ser considerados pela função utilidade dos candidatos à liderança, como taxas de transferência, quantidade de nós alcançáveis pelo candidato e quantidade de *pieces* já descarregados pelo candidato e pelos nós do grupo. Taxas de

transferência podem parecer uma boa idéia a princípio pois nós com maiores taxas de transferência são capazes de conseguir descarregar os *pieces* e retransmiti-los para os outros nós do grupo mais rapidamente. Porém todos os nós do grupo estão sujeitos ao mesmo meio sem fio e portanto possuem larguras de banda semelhantes. Além disso, nos casos em que os nós começam a descarregar o arquivo ao mesmo tempo, nenhum deles possui nenhuma informação sobre as suas taxas de transferências ainda.

A quantidade de nós alcançáveis pelo candidato e a quantidade de *pieces* já descarregados porém fornecem uma boa medida do ganho que a formação do grupo trará para a rede. Um líder deve atender à maior quantidade de nós servindo a maior quantidade de *pieces* possível. Portanto a função utilidade também deve levar em consideração a quantidade de *pieces* que os nós do grupo possuem, pois quanto menos *pieces* eles possuem mais *pieces* o candidato pode servir. Os *pieces* que o candidato já possui representam um ganho maior pois ele não precisará descarregá-los para transmiti-los para o grupo.

Seja P o conjunto de *pieces* de um arquivo. A função utilidade de um candidato é dada por

$$\sum_{i \in P} n_i \gamma_i$$

em que n_i é a quantidade de nós do grupo, exceto pelo candidato, que não possuem o *piece* i e $\gamma_i = \alpha$ caso este candidato possua o *piece* i e $\gamma_i = \beta$ caso contrário. É razoável afirmar que $\alpha = 2\beta$ pois o segundo caso envolve o dobro de transmissões do primeiro, uma para descarregar o *piece* e outra para enviar para o grupo, enquanto no primeiro caso só é necessário enviar o *piece* para o grupo. Assim, essa função também leva em consideração a quantidade de tráfego local necessária para transmitir um *piece*.

5.2. Transmissão do arquivo

Quando um nó é eleito líder, ele começa a transmitir o arquivo para os nós do grupo. Estes se desconectam da rede P2P para diminuir o tráfego local e aumentar as taxas de transferência do líder. O líder transfere primeiro os *pieces* mais raros dentro do grupo. Esta abordagem garante maior eficiência no início da transferência, assim, no caso do grupo se desfazer antes dos nós completarem as transferências, ele terá transferido para o grupo a maior quantidade de dados que podia. Enquanto o líder envia o arquivo para os nós do grupo, ele descarrega os *pieces* que não tem utilizando a política tradicional do BitTorrent. Se ele decide descarregar um *piece* que um nó do grupo já possui, ele descarrega o *piece* deste nó.

Para um nó que estiver prestes a completar a transferência, a formação de um grupo não é vantajosa para ele pois a chance de os *pieces* escolhidos para serem transmitidos pelo líder não serem os *pieces* que estão faltando para este nó são altas. Assim, ele pode demorar muito mais para terminar a transferência ao se juntar a um grupo do que se continuar descarregando o arquivo sozinho. A entrada de uma semente para o grupo não é vantajosa para a rede P2P pois as sementes realizam um papel importante disseminando o arquivo pela rede e a sua entrada no grupo implica que ela deverá parar de se comunicar com nós fora do grupo. Por outro lado a participação destes nós é vantajosa para o grupo pois o líder pode descarregar os *pieces* que estes nós possuem ao invés de requisitar a um participante de fora do grupo, obtendo-o mais rapidamente. Assim, caso um nó esteja

concluindo uma transferência ou seja uma semente, ele entra para o grupo, mas não deixa de participar da rede P2P.

5.2.1. Confiabilidade da transmissão

Um grande desafio neste cenário é realizar a transmissão do arquivo de forma confiável utilizando mensagens em modo *multicast*. Os mecanismos de retransmissão ou confirmação de mensagens em modo *unicast* não funcionam em transmissões *multicast* porque o número de receptores é maior [Tourrilhes 1998]. Contudo, os mecanismos propostos na literatura para resolver este problema [Lee and Cho 2003, You et al. 2006] ou são ineficientes ou muito difíceis de se implementar.

O meio sem fio adiciona vários desafios à transmissão confiável utilizando mensagens modo *multicast* e *broadcast*. Como não é possível monitorar a ocorrência de colisões, uma vez que elas podem acontecer fora do alcance do rádio do remetente, não é possível utilizar um mecanismo CSMA/CD. Também não é possível utilizar esquemas CSMA/CA tradicionais pois não é possível realizar a negociação pelo meio através das mensagens RTS e CTS. O remetente não sabe a quantidade de receptores de uma mensagem e por consequência quantas mensagens CTS esperar. Além disso, mais de um nó enviando mensagens CTS ao mesmo tempo levaria à ocorrência de colisões [You et al. 2006].

Mecanismos baseados em TDMA oferecem uma alternativa para resolver este problema. A idéia destes sistemas é reservar para cada nó um período de tempo durante o qual os outros nós concordam em não enviar nada, deixando o canal livre para este nó. Esse mecanismo porém exige a sincronização dos nós e não funciona bem em ambientes onde existe mobilidade devido à dificuldade de negociação dos intervalos de tempo quando a topologia muda constantemente [Marina et al. 2001].

Entretanto não é necessário garantir confiabilidade na entrega de cada mensagem. É possível utilizar um esquema que aumente a confiabilidade da transmissão e, nos casos em que uma mensagem não seja entregue corretamente, utilizar um canal *unicast* que embora seja menos eficiente, é confiável. O objetivo é tornar as transmissões mais confiáveis sem penalizar a eficiência do sistema. Além disso, este mecanismo deve ser simples para não desmotivar a sua implementação.

Utilizamos o mecanismo baseado em CSMA/CA proposto por [Eger et al. 2007] para aumentar a confiabilidade das transmissões em modo *multicast*. Escolhemos um mecanismo baseado em CSMA/CA por ser o mesmo mecanismo utilizado pelo protocolo de acesso ao meio IEEE 802.11, que é o mais utilizado por dispositivos móveis atualmente. Além disso, ele é simples de se implementar e exige modificações apenas na camada de acesso ao meio.

Este mecanismo negocia o acesso ao meio trocando mensagens RTS e CTS, mas apenas com um dos destinatários. Isto não elimina a possibilidade de haver terminais escondidos para todos os destinatários, apenas para o destinatário que negocia o acesso ao meio. Contudo, como os nós estão próximos, há uma tendência que a mensagem CTS deste destinatário também alcance os nós escondidos dos demais destinatários, tornando o canal livre para todos os nós do grupo.

Na camada de transporte, não é possível utilizar TCP pois há muitos destinatá-

rios para a mesma mensagem. Dessa forma, optamos por utilizar UDP, que não possui confirmação ou retransmissão. Para garantir confiabilidade a nível de transporte, sempre que um *piece* é completamente enviado pelo líder, ele pergunta, através de uma conexão TCP confiável a cada nó do grupo se há algum pedaço do *piece* que ele não recebeu corretamente. Caso exista, ele o envia a este nó através de uma conexão TCP confiável.

6. Avaliação

Nesta seção, analisamos o comportamento de nós BitTorrent com e sem localidade espaço-temporal quanto ao tempo para completar as transferências e taxas de recepção a fim de evidenciar os problemas enfrentados por essa rede diante desta propriedade. A eficácia da solução proposta neste trabalho é verificada analisando como a nossa extensão do BitTorrent melhora o seu desempenho. O tráfego gerado na rede pela implementação padrão e pela estendida também são analisados para mostrar que a nossa solução diminui a quantidade de tráfego gerada na rede explorando as relações entre nós vizinhos.

6.1. Métricas

Para evidenciar os problemas enfrentados pelo BitTorrent diante de localidade espaço-temporal, é necessário entender o comportamento dos nós que apresentam esta propriedade relativamente ao de nós que não a apresentam. As métricas adotadas são o tempo de transferência médio de um arquivo e as taxas de recepção ao longo do tempo. Ao analisar esta última, é possível verificar como os nós que apresentam localidade espaço-temporal possuem dificuldades em competir pela transferência de arquivos com os demais nós, que descarregam arquivos com facilidade.

Verificamos o desempenho da solução proposta analisando as mesmas métricas descritas acima utilizando a nossa implementação estendida. Além disso, comparamos a quantidade de dados descarregada de nós distantes por nós com localidade espaço-temporal utilizando as duas implementações. Quanto maior a quantidade de dados descarregada de nós distantes, menos os nós com localidade espaço-temporal interagem e maior o tráfego gerado por eles ao longo da rede.

6.2. Método de avaliação

Utilizamos simulações para realizar a avaliação. O simulador escolhido foi o NS-2, que é muito utilizado pela comunidade científica para o estudo de problemas semelhantes ao nosso e que possui vários protocolos implementados. Utilizamos a implementação do BitTorrent para NS-2 proposta em [Eger et al. 2007], que é a única de nosso conhecimento.

O cenário definido para o estudo retrata uma situação de incêndio em um prédio, onde o corpo de bombeiros se prepara para iniciar o trabalho. Esse é um cenário provável na aplicação deste trabalho. Além disso, utilizando este cenário, é possível avaliar as métricas apresentadas na seção 6.1. Cada bombeiro carrega um PDA onde pode visualizar a planta do prédio com informações coletadas por sensores sem fio. Para isso, eles devem descarregar um arquivo de 2MB que contém essas informações utilizando BitTorrent.

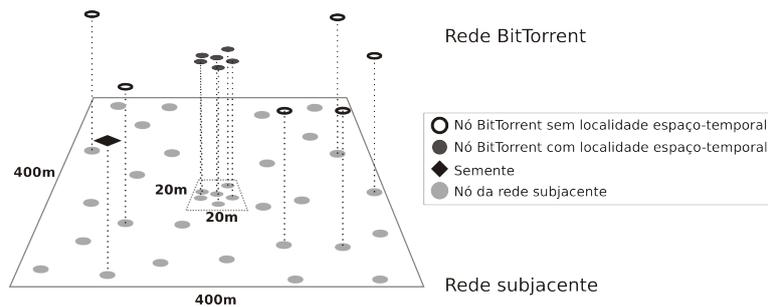


Figura 1. Cenário de estudo

Os bombeiros trabalhando na emergência estão dispostos em uma região geográfica de $400m \times 400m$. Uma equipe de n bombeiros está concentrada em uma área de $20m \times 20m$ se preparando para entrar nas instalações em chamas. Cada um deles está descarregando a planta do prédio utilizando BitTorrent. Por estarem próximos e descarregando o mesmo arquivo ao mesmo tempo, os PDAs desses bombeiros apresentam localidade espaço-temporal. Para comparar o desempenho de nós com e sem localidade espaço-temporal, espalhamos outros n bombeiros descarregando o mesmo arquivo em posições aleatórias do cenário de forma que os PDAs deles não apresentem localidade espaço-temporal. Inicialmente, apenas um nó semente, localizado em uma posição aleatória da região, possui o arquivo. Todos os nós iniciam a transferência ao mesmo tempo. Assim é possível comparar diretamente o desempenho deles. A rede subjacente é formada por 20 nós atuando apenas como roteadores, além dos PDAs dos bombeiros e a semente, que também realizam roteamento. Os nós utilizam rádios IEEE 802.11b com alcance de 40m e o protocolo de roteamento é o DSDV [Perkins and Bhagwat 1994]. Este cenário está ilustrado na Figura 1.

Realizamos 10 rodadas de simulações, variando o valor de n de 6 a 12 a cada rodada, e analisamos a média dos resultados obtidos. Estes são tamanhos razoáveis para equipes de bombeiros. Cada simulação é executada até todos os nós terminarem a transferência.

Quando utilizada a implementação estendida do BitTorrent, após enviar uma mensagem de pesquisa por vizinhos, cada nó espera $t_g = 0,5s$ por mensagens de resposta e, durante a eleição, cada nó espera $t_e = 0,5s$ por submissões de outros nós após enviar a sua função de utilidade. Acreditamos que esses tempos sejam razoáveis pois em nossas simulações, todos os grupos foram formados em no máximo $0,3s$ e as eleições duraram no máximo $0,2s$.

6.3. Resultados e análise

A Figura 2 mostra o tempo médio para completar as transferências. No caso da implementação padrão, os nós sem localidade espaço-temporal apresentam tempos menores que os de nós que possuem esta característica. Isso mostra que o desempenho dos nós BitTorrent é prejudicado sob localidade espaço-temporal. Além disso, o tempo de transferência dos nós com localidade espaço-temporal cresce de forma mais acentuada à medida que aumentamos o valor de n .

Ao utilizar a implementação estendida, foi possível reduzir o tempo dos nós com localidade espaço-temporal tornando os tempos de transferência menores que os dos ou-

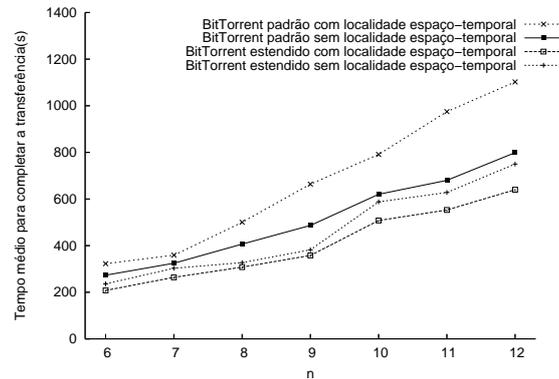


Figura 2. Tempo médio para completar as transferências.

tros nós. Além disso, os tempos de transferência dos nós sem localidade espaço-temporal também foram reduzidos pois a criação do grupo diminuiu o tráfego na rede e sobrecarregou menos a semente, que pode então ceder mais tráfego a estes nós.

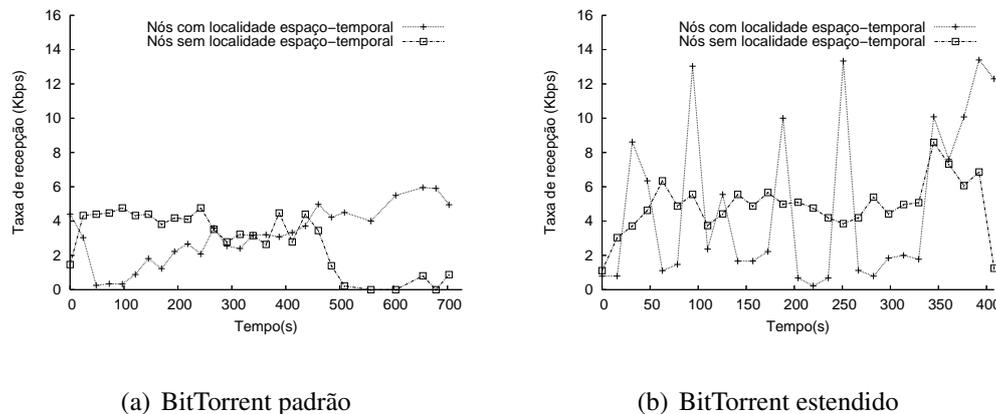


Figura 3. Taxa de recepção média ao longo do tempo com $n = 9$

A taxa de recepção média dos nós com e sem localidade espaço-temporal para o cenário onde $n = 9$ é mostrada na Figura 3. Note que o intervalo de tempo da figura 3(b) é menor que o da figura 3(a) pois os nós completam a transferência do arquivo mais rapidamente quando utilizam a implementação estendida. Na implementação padrão, os nós que apresentam localidade espaço-temporal não são capazes de obter boas taxas de transferência no início da transmissão. Isto se deve ao problema da redução da competitividade por redução das taxas de transferência. Os nós com localidade espaço-temporal não conseguem competir com os nós sem localidade espaço-temporal e só conseguem taxas de recepção significativas à medida que os nós sem localidade espaço-temporal concluem suas transferências, diminuindo a concorrência por transferências.

A implementação estendida é capaz de garantir a mesma competitividade para os nós que apresentam localidade espaço-temporal eliminando este problema. O tráfego

dos nós do grupo segue o perfil de rajadas apresentando vales quando apenas o líder está descarregando *pieces* do resto da rede e picos quando ele termina a transferência de um *piece* e então o envia para os demais nós do grupo, conseguindo altas taxas de transferência graças às transmissões em modo *multicast*.

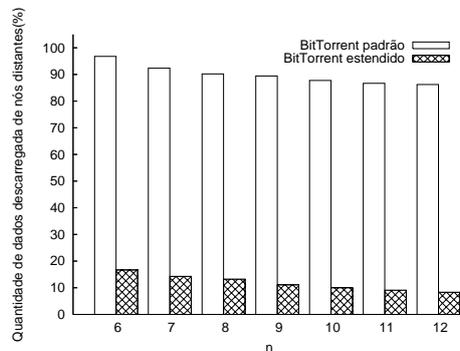


Figura 4. Quantidade relativa de bytes transferidos de nós distantes

A Figura 4 mostra a quantidade de dados que é descarregada de nós distantes. Na implementação padrão, os nós com localidade espaço-temporal descarregam quase todo o arquivo de nós distantes embora seja preferível interagir com os nós próximos, sobrecarregando menos a rede. Esse resultado é uma consequência do problema da redução da competitividade por redução das taxas de transferência. Como os nós distantes, que não apresentam localidade espaço-temporal, descarregam o arquivo a taxas mais elevadas, eles possuem mais *pieces* que os nós com localidade espaço-temporal. Assim a probabilidade de conseguir *pieces* em nós distantes é maior do que em nós próximos. A implementação estendida consegue reduzir este valor drasticamente fazendo com que o arquivo seja descarregado apenas uma vez pelo líder e então redistribuído localmente.

7. Conclusão e trabalhos futuros

Neste trabalho estudamos a questão de disseminação de dados em redes críticas utilizando redes P2P. O BitTorrent possui propriedades que fazem dele uma rede P2P ideal para este propósito, porém não é capaz de lidar com localidade espaço-temporal, uma propriedade presente em redes críticas, e por isso os nós que apresentam esta propriedade encontram dificuldades em descarregar arquivos e utilizam os recursos da rede de forma ineficiente. Propusemos uma modificação *cross-layer* do BitTorrent para resolver estes problemas. Esta solução se mostrou capaz de elevar as taxas de transferência destes nós e de aumentar a interação entre eles, diminuindo o tráfego pelo resto da rede.

Ainda há margem para trabalhos futuros. Não tratamos a questão da mobilidade, que impõe uma série de desafios a esse problema. O problema de confiabilidade pode ser melhor explorado e o conceito de grupo pode ser estendido para acomodar nós a mais de um salto de distância. A avaliação deste sistema em dispositivos reais traria resultados mais consistentes e confiáveis.

Agradecimentos

Agradecemos as valiosas contribuições de Rande Arievidlo, Diego Neves, Henrique Chevreux, Flávio Andrade, Cristiano Nascimento e do Professor Dorgival Guedes.

Referências

- Cohen, B. (2003). Incentives build robustness in bittorrent. In *1st Workshop on Economics of Peer-to-Peer Systems*.
- Conti, M., Gregori, E., and Turi, G. (2005). A cross-layer optimization of gnutella for mobile ad hoc networks. In *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 343–354.
- da Hora, D. N., Macedo, D. F., Nogueira, J. M., and Pujolle, G. (2007). Otimizando requisições de conteúdo em redes par-a-par sobre redes ad hoc. In *24° Simpósio Brasileiro de Redes de Computadores*.
- Delmastro, F. (2005). From pastry to crossroad: Cross-layer ring overlay for ad hoc networks. In *Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 60–64.
- Eger, K., Hoßfeld, T., Binzenhöfer, A., and Kunzmann, G. (2007). Efficient simulation of large-scale p2p networks: Packet-level vs. flow-level simulations. In *2nd Workshop on the Use of P2P, GRID and Agents for the Development of Content Networks (UPGRADE-CN'07)*, pages 9–16, Monterey Bay, USA.
- Lee, K.-H. and Cho, D.-H. (2003). A multiple access collision avoidance protocol for multicast service in mobile ad hoc networks. In *The 57th IEEE Semiannual Vehicular Technology Conference*, pages 1793–1797, Jeju, Korea.
- Marina, M. K., Kondylis, G. D., and Kozat, U. C. (2001). Rbrp: a robust broadcast reservation protocol for mobile ad hoc networks. In *IEEE International Conference on Communications*, pages 878–885, Helsinki, Finland.
- Oliveira, L. B., Siqueira, I., Macedo, D. F., Loureiro, A. A., and Nogueira, J. M. (2005). Evaluation of peer-to-peer network content discovery techniques over mobile ad hoc networks. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 51–56.
- Perkins, C. E. and Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers. *SIGCOMM Comput. Commun. Rev.*, 24(4):234–244.
- Rajagopalan, S. and Shen, C. (2006). A cross-layer decentralized bittorrent for mobile ad hoc networks. In *The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services*, pages 1–10.
- Tourrilhes, J. (1998). Robust broadcast: Improving the reliability of broadcast transmissions on csma/ca. In *The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, pages 1111–1115.
- Y. Charlie Hu, S. M. D. and Pucha, H. (2003). Exploiting the synergy between peer-to-peer and mobile ad hoc networks. In *Proceedings of HotOS IX: The 9th Workshop on Hot Topics in Operating Systems*.
- You, T., Hassanein, H., and Yeh, C.-H. (2006). Semac: robust broadcast mac protocol for multi-hop wireless ad hoc networks. In *The 25th IEEE International Performance, Computing, and Communications Conference*, pages 71–78, Phoenix, Arizona.